昭和63年8月1日発行（毎月1回1日発行）第7巻8号通巻76号　昭和58年11月2日第三種郵便物認可
ISSN 0910-7614

特集1
真夏の夜の数値演算

永遠不滅のπ/高速FLOAT 3+.X
サウンド&グラフィックへの応用

特集2
MIDIサウンドプログラミング

MIDI対応シーケンサ発表
再掲載MIDIボードの製作

新連載 Z80マシン語ゲーム工房
目指せシューティング

C調言語講座 PRO-68K
ほいほいファイル術

S-OS 全機種共通システム
マルチウィンドウエディタWINER

ミュージックプログラム LIVE in '88
MZ-2500ささやきのステップ/X1/turboマリオネット
組曲「くるみ割り人形」よりシナの踊り

THE SOFTOUCH
イースII/ソーサリアン/第4のユニット2

新連載　われら電脳遊戯民

X68000 BASIC入門/あなたの知らない世界
オブジェクト指向のゲームプログラミング
猫とコンピュータ

8
AUG.1988
定価540円

SHARP

■本体＋キーボード＋マウス・トラックボール
CZ-611C-GY(グレー)・-BK(ブラック)標準価格399,800円

■本体＋キーボード＋マウス・トラックボール
CZ-601C-GY(グレー)・-BK(ブラック)標準価格319,800円

〈X68000ACEシリーズの主な特長〉●実装密度をさらに追求して信頼性を高めたマンハッタンシェイプ●68000搭載●テキスト、グラフィック、スプライト、独立3画面設計、最大12Mバイトの大容量メモリ(標準1Mバイト)●フレンドリーOS、Human 68k搭載●連文節変換、マルチフォントをサポートした強力日本語処理●1024×1024ドット(最大表示エリア768×512ドット)の実画面エリアを装備した高解像度表示能力●512×512ドット、65,536色同時発色●水平32、1画面128のパワフルなスプライト機能●オーバースキャン機能を採用した512×512ドットレベルのスーパーインポーズ●テキストビットマップ方式採用●8重和音ステレオFM音源搭載●音声デジタイズ記録AD PCM※採用●マウス・トラックボール標準装備●5インチ1MバイトFDD2基搭載●「X-BASIC」、「辞書ディスク」と各種ユーティリティ、「日本語ワードプロセッサ」をバンドル●20Mバイトハードディスク内蔵(CZ-611C)

※Adaptive Differential PCM

■15型カラーディスプレイテレビ(ドットピッチ0.39mm)CZ-601D-GY(グレー)・-BK(ブラック)標準価格119,800円
■15型カラーディスプレイテレビ(ドットピッチ0.31mm)CZ-611D-GY(グレー)・-BK(ブラック)標準価格145,000円
■チルトスタンドCZ-6ST1-E(グレー)・-B(ブラック)標準価格5,800円

資料請求券 X68000

アートの領域へ。

クォリティを維持しつづけることは、ある意味では創造することより困難なこととも言われています。出会いが印象的であればあるほど、その後が大変です。このことは、そのままX68000の歩みを言い得ているかも知れません。確かに技術は日進月歩です。しかしそれだけでコンピュータがもつべき創造性を論ずることはできないのも、また事実です。私たちはテクノロジーとクリエイティブマインド、いわば人とマシンとのソフトウェアインターフェイスで応えます。ホリゾンタルなマシンとしての熟成。そこからはいくつもの分野が見えてくるはずです。そしてどんな分野にしろX68000の仕事はアートであるべきです――。ますます洗練されて信頼性を高めたACEシリーズの登場で、あなたはまた新たな可能性に出会えそうです。

豊富な周辺機器がクリエイティブワークをサポート。

品名	型名	価格
●15型カラーディスプレイ	CU-15M1-E	標準価格 99,800円
●カラーイメージスキャナ※1	CZ-8NS1	標準価格 188,000円
●カラーイメージユニット※2	CZ-6VT1	標準価格 69,800円
●カラービデオプリンタ	CZ-6PV1	標準価格 198,000円
●24ピン漢字プリンタ(80桁)	CZ-8PK7	標準価格 122,000円
●24ピン漢字プリンタ(136桁)	CZ-8PK8	標準価格 152,000円
●24ピン漢字プリンタ(80桁)	CZ-8PK9	標準価格 89,800円
●熱転写カラー漢字プリンタ	CZ-8PC2	標準価格 69,800円
●ハードディスクユニット(20MB)	CZ-620H	標準価格 178,000円
●モデムユニット※3	CZ-8TM2	標準価格 49,800円
●RS-232Cケーブル(平行接続型)	CZ-8LM1	標準価格 7,200円
●RS-232Cケーブル(クロス接続型)	CZ-8LM2	標準価格 7,200円
●拡張I/Oボックス(4スロット)	CZ-6EB1	標準価格 88,000円
●1MB増設RAMボード(内蔵用)	CZ-6BE1A	標準価格 38,000円
●2MB増設RAMボード※4	CZ-6BE2	標準価格 79,800円
●4MB増設RAMボード※4	CZ-6BE4	標準価格 138,000円
●FAXボード	CZ-6BC1	標準価格 79,800円
●GP-IBボード	CZ-6BG1	標準価格 59,800円
●ユニバーサルI/Oボード	CZ-6BU1	標準価格 39,800円
●増設用RS-232Cボード(2チャンネル)	CZ-6BF1	標準価格 49,800円
●数値演算プロセッサボード	CZ-6BP1	標準価格 79,800円
●スキャナ用パラレルボード	CZ-6BN1	標準価格 29,800円
●システムラック	CZ-6SD1	標準価格 44,800円
●アンプ内蔵スピーカーシステム(2本1組)	AN-160SP	標準価格 59,800円
●トラックボール	CZ-8NT1	9月発売予定
●ジョイカード	CZ-8NJ1	標準価格 1,700円

※1 使用に際しては、カラーイメージスキャナCZ-8NS1に同梱のRS-232Cケーブルで接続するか、より高速のパラレルデータ転送を行う場合、別売のスキャナ用パラレルボードCZ-6BN1で接続してください。※2 使用に際してはコンピュータ本体と専用15型カラーディスプレイテレビ(CZ-601D、CZ-611Dなど)が必要です。※3 モデムユニットCZ-8TM2に同梱のソフトはX1/X1 turboシリーズ用です。※4 使用に際しては、あらかじめ、別売の1MB増設RAMボードCZ-6BE1Aを増設してください。

アートツールと呼びたい「PRO-68K」シリーズソフト。

ソフト	型名	価格
イージーオペレーションの統合型表計算ソフト BUSINESS PRO-68K	CZ-212BS	標準価格 68,000円
コマンド型リレーショナルデータベース DATA PRO-68K	CZ-220BS	標準価格 58,000円
ワープロ機能を備えたカード型リレーショナルデータベース CARD PRO-68K	CZ-226BS	標準価格 29,800円
FM音源をフルサポートするサウンドエディタ SOUND PRO-68K	CZ-214MS	標準価格 15,800円
マウスを使った簡単操作の楽譜ワープロ MUSIC PRO-68K	CZ-213MS	標準価格 18,800円
AD PCM機能をサポートしたサンプリングエディタ Sampling PRO-68K	CZ-215MS	標準価格 17,800円
オリジナリティを活かせるポップアートツール NEW Print Shop PRO-68K	CZ-221HS	標準価格 19,800円
フルスクリーンエディタ内蔵の通信ソフト Communication PRO-68K	CZ-223CS	標準価格 19,800円
ソフトウェア開発に役立つCコンパイラ C compiler PRO-68K	CZ-211LS	標準価格 39,800円
ソフトウェア開発ツール THE 福袋 V2.0	CZ-224LS	標準価格 9,980円
マルチタスク、リアルタイムオペレーティングシステム OS-9/X68000		10月発売予定

〈ゲームソフト〉

タイトル	型名	価格
●ツインビー	CZ-217AS	標準価格 7,800円
●アルカノイド	CZ-222AS	標準価格 7,800円
●沙羅曼蛇		9月発売予定
●熱血高校ドッジボール部		10月発売予定
●フルスロットル		8月発売予定

〈パソコン教室開催のお知らせ〉 X68000、MZ-2861のパソコン教室を開催します。くわしくは、下記までお問い合せください。

札幌(011)642-8111・仙台(022)288-8705・東京(03)260-1161・横浜(045)201-6525・名古屋(052)332-2611・大阪(06)222-7655・神戸(078)291-8715・福岡(092)481-2860

シャープ株式会社 ●お問い合わせは…シャープ㈱電子機器事業本部システム機器営業部 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)
電子機器事業本部テレビ事業部第4商品企画部 〒162 東京都新宿区市谷八幡町8番地 ☎(03)260-1161(大代表)

表紙絵：Matsubaguchi Tadao

Oh!X

CONT

●特集1

28 **真夏の夜の数値演算**

28 コンピュータにおける数値表現 村田敏幸

34 連立方程式は行列でいこう 相馬英智

42 *i*があるからむずかしい 向原あゆむ

47 とんでもなくデタラメな話 華門真人

51 そこにπがあるから 丹 明彦

61 超応用グラフィック歪められた光 枀野雅彦

64 超応用AD PCM音の数学 加藤賢哉

72 数値演算プロセッサの活用FLOAT3+.X 長井 清/平野照比古/中野修一

●特集2

100 **MIDIサウンドプログラミング**

100 短期集中講座 MIDI活用テクニック MIDIの基礎とボードの製作 三沢和彦

106 MIDI対応シーケンサ 金子俊一

●シリーズ全機種共通システム

137 THE SENTINEL

138 マルチウィンドウエディタWINER 近藤 環

〈スタッフ〉
●編集長／前田 徹 ●副編集長／永野 仁 ●編集／植木章夫 石塚康世 高野庸一 ●協力／有田隆也 中森 章 清水和人 後藤貴行 林 一樹 浅野恵造 山村 一 井本 泰 堀内保秀 荻窪 圭 藤原和典 岡本浩一郎 毛内俊行 野中俊一郎 吉田賢司 影山裕昭 相馬英智 古村 聡 村田敏幸 倉持亮一 ●カメラ／杉山和美 ●イラスト／永沢しげる 山田晴久 小栗由香 ●アートディレクター／島村勝頼 ●レイアウト／元木昌子 AD GREEN ●校正／手塚喜美子 千野延明

■広告目次

アイピーエル……186・187
アイビット電子……182・183
アクセス……192
イースト……9
AVCフタバ電機……178
エス・ピー・エス……10
計測技研……176
サムシンググッド……174
J&P……表3・188-191
シャープ……表2・表4・1・4-7
ソフトクリエイト……177
九十九電機……13
日本ファルコム……11
パシフィックコンピュータバンク…184・185
パソコンショップハドソン……8
ピー・アンド・エー……180・181
マイクロネット……12
満開製作所……174
メディアショップ・ハイランド……179

1988AUG.
8

E N T S

●THE SOFTOUCH

14 SOFTWARE INFORMATION
話題のソフトウェア/新作ソフト情報

16 GAME REVIEW
ヘルツォーク/プロフェッショナル麻雀悟空/ハーレッシュ

SPECIAL REVIEW
18 イースII(第1話) 華門真人
20 ソーサリアン(その2) 西川善司
22 第4のユニット2 長沢淳博

24 われら電脳遊戯民(1)
ゲームは僕らのキャンバスだ 倉持亮一

●連載/紹介/システム

26 猫とコンピュータ 第26回
ぼくはかぐや姫？ 高沢恭子

80 Oh! X LIVE in '88
組曲「くるみ割り人形」よりシナの踊り(X1/X1turbo) 伊藤圭一
マリオネット(X1/X1turbo) 佐々木孝司
ささやきのステップ(MZ-2500) 岡上圭作

85 C調言語講座 PRO-68K 第2回
ほいほいファイル術 祝 一平

89 特別講義
Cとアセンブリ言語をリンクして使う 中森 章

96 X68000あなたの知らない世界
CONCERTO-X68K
DATA PRO-68K/CARD PRO-68K

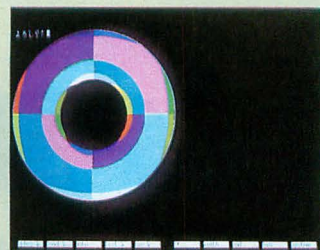
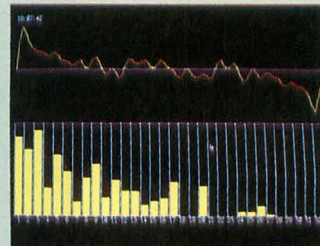
117 Z80マシン語ゲーム工房 第1回
目指せシューティング 村田敏幸

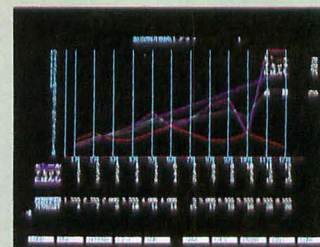
125 実用(?)オブジェクト指向のゲームプログラミング 最終回
オブジェクト指向のゆくえ 浜口 勇

129 X68000 BASIC入門 最終回
必殺サンプリング戦法 中森 章

Oh! X質問箱……156
FILES Oh! X……158
バックナンバー案内……160
ペンギン情報コーナー/Again Watch……161
STUDIO X……164
愛読者プレゼント……168
編集室から/DRIVE ON/ごめんなさいのコーナー/SHIFT BREAK/microOdyssay……170

特集I 歪められた光

特集I 音の数学

イースII

第4のユニット2

X68000あなたの知らない世界

マルチウィンドウエディタWINER

パソコンフリークたちへ

パソコンとしての確かな伝統

コンパチブル設計

X1シリーズの高機能を継承したコンパチブル設計、蓄積された豊富なソフトウェア資産*が利用できます。※カセットテープソフトは利用できません。

●伝統を受け継いだ多彩なグラフィック機能やスーパーインポーズ機能、サウンド機能●JIS第1水準準拠漢字ROM内蔵(漢字ユーティリティソフト付)●5″FDD1基内蔵、別売のCZ-53F(標準価格19,800円)の増設でデュアルドライブも可能●ユーザー定義のキャラクタゼネレータ機能

●CZ-53F

マルチビジュアル端子

コンピュータ画面をビデオ録画できる――。ビデオやビデオ入力端子つきテレビとダイレクトに接続、マルチビジュアル端子がパソコンシーンを鮮やかに彩ります。たとえばゲーム、プレイしながらその過程をそのまま録画、後で再生すれば攻略法も研究できるし、隠れキャラクタやウラ技も確認できる。またベストスコアの達成や最終面をクリアした決定的瞬間もバッチリ残せます。

X1アミューズメントステーション

どちらから始めるか。ニューエンター

HEシステム搭載

リアルなキャラクタで迫力あふれるゲームが楽しめるホームエンターテイメントシステムをX1に搭載しました。HEシステム専用カスタムCPUや高機能多色化スプライトIC、6重和音のサウンド機能、さらにマルチビジュアル端子接続による鮮明画像、ソフトはコンパクトな専用ICカード。この新しさがオモシロさ、もう遊び心はトップギア…。次世代ゲームが思いっきり楽しめます。

■鮮やかな画像/マルチビジュアル端子による鮮明画像。ゲームプレイをビデオに録画もOK。

■リアルなキャラクタ/最大32×64ドットの大迫力キャラクタで、よりリアルなゲームプレイ。

■多彩なカラー表現/表示色は512色中256色同時表示、キメ細かな色彩で表現力がさらにアップ。

■迫るサウンド/6オクターブ6重和音のサウンド機能でさらにひろがる臨場感。

■ICカード/ソフトは手のひらに入る専用ICカード、遊び心が一気に加速する新しさ。

次世代ゲームマシンの高感度

ゲームフリークたちへ

これがX1誕生5年目の解答です。

(システムアップも)

サウンド、アート、通信も…。これは成長する楽しみ。テレビやビデオの映像をカラー静止画で瞬時に取り込めるカラーイメージボード※1、ステレオタイプのFM音源※2、話題のネットワークにアクセスしたり、仲間同士でデータやメッセージを交換できるパソコン通信※3もサポートします。

※1 カラーイメージボードII CZ-8BV2 標準価格 39,800円
熱転写カラー漢字プリンタ CZ-8PC2 標準価格 69,800円
※2 ステレオタイプFM音源ボード(スピーカー2本1組標準装備・ミュージックツール同梱) CZ-8BS1 標準価格 23,800円
※3 モデムユニット(300ボー) CZ-8TM1 標準価格 29,800円・モデムユニット(300ボー/1200ボー自動切換) CZ-8TM2 標準価格 49,800円

テイメントマシン 登場。

専用パッド

■専用パッド/HEシステム専用のパッドを同梱、思いっきりゲームに熱中。

このマークはホーム エンターテイメントシステムの意味です。X1twinのHEシステム用ソフトには、このマークのついているICカードをご使用ください。

●ソフトはコンパクトな専用ICカード

新登場

パソコンテレビ
X1 twin

●パーソナルコンピュータ+キーボード	CZ-830C-BK(ブラック)	標準価格 99,800円
●14型カラーディスプレイテレビ	CZ-830D-BK(ブラック)	標準価格 98,000円
●チルトスタンド	CZ-6ST1-B(ブラック)	標準価格 5,800円

シャープ株式会社

●お問い合わせは…シャープ㈱電子機器事業本部システム機器営業部 〒545 大阪市阿倍野区長池町22番22号 ☎(06)621-1221(大代表)
電子機器事業本部テレビ事業部第4商品企画部 〒162 東京都新宿区市谷八幡町8番地 ☎(03)260-1161(大代表)へ。

資料請求券
X1 twin
oh! X
8係

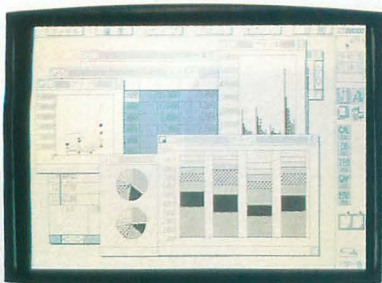
SHARP

ハードの余裕がフレンドリーなオペレーションを生みだしている。インテリジェントな機能に

イージーオペレーションの統合型表計算ソフト

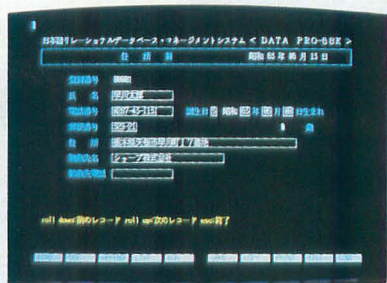
BUSINESS PRO-68K

■CZ-212BS 標準価格68,000円

スプレッドシート(表計算)、データベース、グラフ作成機能を緊密に一体化させた統合ビジネスツールです。マウス対応のやさしいオペレーション、最大16個のマルチウインドウ、高度なエディタ機能、豊富な関数群など、初心者からプロフェッショナルまで幅広くお使いいただけるソフト。定型業務、各シミュレーションにも対応できるよう集計、再計算もスピーディです。

コマンド型リレーショナルデータベース

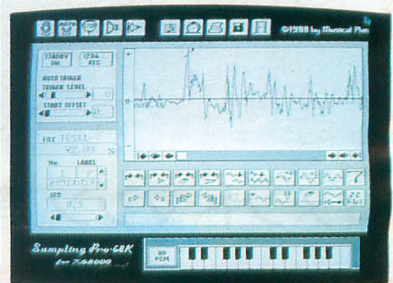
DATA PRO-68K

■CZ-220BS 標準価格58,000円

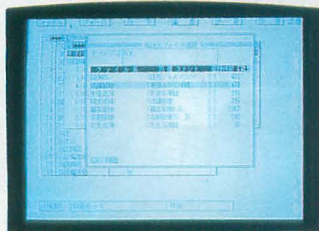
コマンド入力を軽減するヒストリー機能、罫線ドライバー付レポートライター機能、10進31桁の高度な演算精度。またコマンド型RDBとしては初のイメージ表示機能を装備、イメージスキャナ等で取り込んだ絵や写真のデータも管理できます。強力なADL(専用言語)も装備。高度なアプリケーションの構築が可能。さらにサウンドデータの処理もできます。

AD PCM機能をサポートしたサンプリングエディタ

Sampling PRO-68K

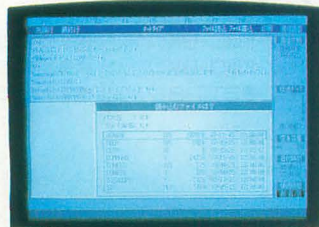
■CZ-215MS 標準価格17,800円

X68000のAD PCM機能を活かすエディタ。録音した音声を波形表示し、それをエディットできるWAVE EDITOR、録音した50音データでX68000がしゃべるSPEACH EDITORなどをサポート。また短いサンプリング音を長く伸ばして持続音が作れるループ処理機能も装備。サンプリングしたデータはBASICプログラムでも使用可能。効果音やおしゃべりを多彩に活かせます。

カード型リレーショナルデータベース

CARD PRO-68K

■CZ-226BS 標準価格29,800円

スクリーンエディタ内蔵の高機能通信ソフト

Communication PRO-68K

■CZ-223CS 標準価格19,800円

マウスを使った簡単操作の楽譜ワープロ

MUSIC PRO-68K

■CZ-213MS 標準価格18,800円

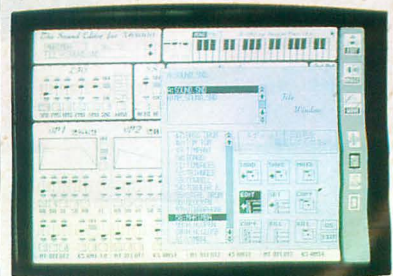
メロディ譜、ピアノ譜、最大8パートのスコア(総譜)を自由なレイアウトで書き込んだ譜面を、内蔵のFM音源で演奏できる楽譜ワープロ&演奏用ミュージックツールです。音符データの入力/編集(複写・削除・挿入)はマウスでとても簡単。プルダウンメニューから音符や記号を選んで五線譜に置いていくだけで楽譜が入力できます。「SOUND PRO-68K」との連動も可能。

FM音源をフルサポートするサウンドエディタ

SOUND PRO-68K

■CZ-214MS 標準価格15,800円

まるでスタジオのコンソールパネルを操作する感覚で音作りが楽しめるサウンドエディティングツール。マウスを使ってFM音源のパラメータを直接指定したり、エンベロープやビブラートを音のイメージ、たとえば明るい/暗い、鋭い/やわらかいなど、言葉による指定で思いどおりの音色が作成できます。サンプリングシンセサイザでおなじみの3次元表示するモードも装備。

資料請求券
CZソフト
oh!X
8体

さらに密な環境へ。

クリエイティブマインドあふれるソフトウェアがX68000をサポート。

シャープオリジナルソフトウェア X68000

「PRO」と称される理由がわかる。

オリジナリティを活かせるポップアートツール

NEW Print Shop PRO-68K

■CZ-221HS 標準価格19,800円

オリジナリティあふれるはがき、便せん、グリーティングカード等を簡単に作成、印刷できるホームプロダクティビティツール。ほとんどの処理をアイコンで表示し、マウスで選ぶフレンドリーなビジュアルコントロール。高機能グラフィックエディタも内蔵、データの加工・修正もOK。X-BASIC(img_save)で作成したグラフィックデータやZ'S STAFF PRO-68Kで作成したデータも活用できます。

ソフトウェア開発に役立つCコンパイラ

C compiler PRO-68K

■CZ-211LS 標準価格39,800円

X68000のソフトウェア開発に役立つCコンパイラ(XC)、BASIC-Cコンバータ(X BAStoC)、アセンブラ(X Assembler)、リンカ(X Linker)、デバッガ(X Debugger)、アーカイバ(X Archiver)、コンバータ(X Converter)からなるツール。Human68K上におけるプログラム開発を効率良くサポートします。

●X-BASICのソースプログラムをXCのソースプログラムに変換するBASIC-Cコンバータで、X-BASICによるマシン語開発をサポート。

●XCはC言語の最も基本的な仕様(K&R)に準拠し、ANSI仕様も取り入れた最新バージョン。また標準ライブラリ、日本語ライブラリ、IOCSライブラリ、DOSライブラリ、BASICライブラリなど、ハードウェアをサポートした豊富なライブラリ(約700種)が用意されています。

フルスロットル

ドライブゲーム
8月発売予定

熱血高校ドッジボール部

スポーツゲーム
10月発売予定

ソフトウェア開発ツールセット

THE福袋 V2.0

■CZ-224LS 標準価格 9,980円

マルチタスク、リアルタイムオペレーティングシステム

OS-9/X68000

10月発売予定

X68000の持つ高度なグラフィック環境はもちろん、AD PCM、FM音源とグラフィックスの同時再生処理といったマルチメディアに対応。本格的なオーバーラッピングマルチウインドウがサポートされ、プログラミングレベルからアプリケーションレベルまで使いやすく機能的な環境を提供します。

※OS-9はマイクロウェア社の登録商標です。

ツインビー

シューティングゲーム
■CZ-217AS
標準価格
7,800円

アルカノイド

ブロックゲーム
■CZ-222AS
標準価格
7,800円

沙羅曼蛇

シューティングゲーム
9月発売予定

各システムハウスのアプリケーションも続々登場

●日本語ワープロ

EW(イー・ダブリュー)	38,000円	イースト㈱

●表集計・データベース

ビジレスAD68K	98,000円	MASH SYSTEMS

●統合化ソフト

Kamikaze(神風)	68,000円	㈱サムシンググッド

●グラフィックツール

Z'S STAFF PRO-68K	58,000円	㈲ツァイト
Hyper UD	16,800円	イースト㈱
Hyper UD Com・Viリンク	35,000円	㈱C&B

●通信ソフト

X Link PRO-68K	19,800円	シスポート㈱

●OS

CONCERT-X68K	99,800円	㈲アクセス
CP/M-68Kエミュレータ	19,800円	㈱計測技研
CP/M-68K	110,000円	ニューウェイブ
CP/M-68Kエミュレータ	30,000円	ニューウェイブ

●システムユーティリティ

BASIC拡張関数パッケージ	9,800円	㈱計測技研
アイコンエディター	4,800円	㈱計測技研
ディスクキャッシャー	6,800円	㈱計測技研
WINDEX PRO-68K	28,000円	㈱ジェー・イー・エル
ファイルコンバータ	20,000円	ニューウェイブ
アプリケーションソフト PS-HDD MAKE ver 2.0	9,800円	パソコンショップ ハドソン

●ゲーム

ハウメニ・ロボット	9,500円	㈱アートディンク
魔神宮	7,800円	㈱ザイン・ソフト
グランドマスター	9,800円	㈱ザイン・ソフト
DOME	9,800円	システムサコム
上海	6,500円	㈱システムソフト
スペースハリヤー	6,800円	電波新聞社
T.D.F	6,800円	データウェスト㈱
源平討魔伝	7,800円	電波新聞社
ザ・ラスベガス	9,800円	日本デクスタ㈱
レリクス	7,200円	ボーステック㈱
マンハッタン・レクイエム	7,800円	㈱リバーヒルソフト
桃太郎伝説	7,800円	㈱ハドソン

※この他、既発売、発売予定のソフトが約110本。詳しくはX68000シリーズ用「SOFTWARE FIELD」をご参照下さい。

シャープ株式会社 ●お問い合わせは…シャープ㈱電子機器事業本部テレビ事業部 第4商品企画部 〒162 東京都新宿区市谷八幡町8番地 ☎(03)260-1161(大代表)へ。

FOR SHARP X68000

■マルチな遊・感覚で評判のハイパーUDが〈カラーイメージユニット対応〉〈テロップのスムーズスクロールサポート〉〈EI搭載〉の強力バージョンUPで新登場!!

SHARP X68000対応
¥21,800(E1付)

パソコンが持つグラフィック機能、ミュージック機能、サウンド機能など、これらの独立したマルチ機能を統合したハイパーUD。プログラミングすることなく、絵や音が自由にエディットできるクリエイティブソフトです。パソコン紙芝居、アニメーション、パーソナルゲーム、デスクトッププレゼンテーション、各種教材、さらにビデオ編集に有効に利用できます。

GRAPHICS Editor
ペンやブラシを使って描画を
画面いっぱいにペンやブラシ、スプレーなどを使って絵や文字が自由に描けます。円や四角、直線を書いたり、塗りつぶしも思いのまま。65537色中240色を同時表示可能です。

TELOP Editor
テロップ作成も容易
あらかじめ設定しておいたテロップをシナリオの手順に従って流すことができます。文字サイズ、エッジング、バックカラーの指定は自由。テロップの方向、場所、スピードも選べます。

SPRITE Editor
スプライトでアニメ作成を
32×48ドット、64×96ドットのスプライトが作成できます。人物や動物などのキャラクターをいくつも作成しておいて、これを続けて表示すればアニメーションやパーソナルゲームが作れます。スプライトの表示順序、速度、移動量、移動ルートが決められます。

MUSIC Editor
メロディ、コード、リズム、パターンの設定
画面に表示された鍵盤をマウスで選択するだけの手軽さ。オリジナル曲も簡単に譜面に書き表すことができます。コード、リズム、パターンはもちろん、楽器の種類の設定もできます。

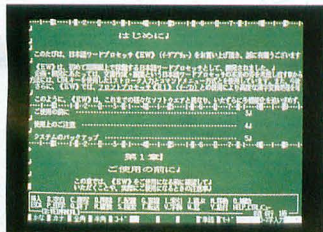
FREE HAND Editor
マウスを使ってタイトル文字を

VIDEO
ホームビデオの編集もOK

SCRIPT
シナリオ(構成)作成も容易

VOICE Editor
ナレーションの録音・再生が可能

■個性が光るクイック・ワープロです。

EW &E1

機能の数を重視する現在の日本語ワープロの中にあっては、EWは非常に個性的です。当たり前のことですが、ワープロ本来の機能と操作性を重視し、シンプルで使いやすいワープロを目指しました。ですから、スクロールなども早いですし、印刷も、わざわざメニューに戻らなくても瞬時に印刷モードに入れる使いやすさです。また、索引や目次の自動作成など、まさに文書作りに徹した個性が光ります。

■マルチウィンドウ画面

■EWの主な特長 ▶強力な印刷機能▶ドキュメント作成に便利な目次、索引の自動作成▶エディタモードの標準サポート▶独自のカナ漢字変換プロセッサE1標準搭載▶他文書参照やカット&ペーストが行なえるマルチウインドウ処理▶編集画面からのOSコマンド及びユーザープログラム実行▶表を含む文章での強力なブロック操作▶MULTIPLANに準拠したコマンドメニュー方式▶WORD MASTERに準拠したコントロールコマンドも容易▶ファイルの大きさに制限のない仮想メモリー方式採用▶バックアップファイルを自動作成する安全設計▶OS上で稼働し標準テキストファイルを生成します。

■目次と索引の自動作成

SHARP X68000対応 ¥38,000(E1付)

イースト株式会社 本社/〒151 東京都渋谷区代々木1-3-1 ☎(03)374-1980(代表)

ソーサリアンは進化する。

Falcom

ソーサリアンユーザーだけが体験できるゲーム世界の幕あけ。新システム登場！

X1版7月29日発売！

ソーサリアン

Socerian System

ソーサリアン システム

SOCERIAN SYSTEM

「ソーサリアン」はRPG初のシステム（MS-DOS・BASIC等）構造の自由度の非常に高い増殖するRPGゲームです。
追加シナリオの発売により、「ソーサリアン」の世界が増殖して行きます。
追加シナリオは舞台、イベント、モンスター、BGM、等の設定が多彩に展開し、これからも続々と発売されます。

同一機種・メディアの「ソーサリアン」が必要です。

新発売

	機　種	メディア	定価	発売日
ソーサリアン追加シナリオ	98F VM VX	5 2DD	3,800	'88・7・22
ソーサリアン追加シナリオ	98U UV UX	3.5 2DD	3,800	'88・7・22
ソーサリアン追加シナリオ	88SRシリーズ	5 2D	3,800	'88・7・29
ソーサリアン追加シナリオ	X1turbo	5 2D	3,800	'88・7・29
ソーサリアン追加シナリオ	88VAシリーズ	5 2DD	3,800	'88・7・29
ソーサリアン・ユーティリティ-DISK	98F VM VX	5 2DD	3,800	'88・7・22
ソーサリアン・ユーティリティ-DISK	98U UV UX	3.5 2DD	3,800	'88・7・22
ソーサリアン・ユーティリティ-DISK	88SRシリーズ	5 2D	3,800	'88・7・29
ソーサリアン・ユーティリティ-DISK	X1turbo	5 2D	3,800	'88・7・29
ソーサリアン・ユーティリティ-DISK	88VAシリーズ	5 2DD	3,800	'88・7・29

『ソーサリアン』追加シナリオ　今後も続々と発売される「ソーサリアン」の追加シナリオでシナリオディスクVol.1 これで「ソーサリアン」のシナリオは全部で20本になります。同一機種・メディアの「ソーサリアン」が必要です。

君はまだクリアしていない　待望の追加シナリオ創刊。

SOCERIAN SYSTEM SCENARIO Vol.1

1. 魔性の島

沖合いのとある名も無き島で奇妙な事件が起きていた。
島のすぐそばを通る船が次々と消息を絶っている。
かつて、海賊の住家だったその島にいったいどのような謎が秘められているのだろうか？

2. いけにえの神殿

とある村の近くの古代遺跡に、一人の女神官がモンスターを従えて現れた。「新月の夜に若い娘をいけにえとしてさしださねば世界を破滅においやる」と言うのだが

3. 悪魔に魅いられた花

数多くの貴金属を出す鉱山の村で、疫病がはやりだしたという情報が流れてきた。村のそばにあるカルテラの山の窪地に良く効く薬草が生えているというのだが、そこへの道は落磐のため閉ざされていた

4. ああ、ジョセフィーヌは今何処に…

ケメケメ王国のプリンセス　マーヘラ　の可愛がっていたペットのジョセフィーヌが、城の地下に広がるダンジョンに迷い込み行方不明になってしまったのだ！？
さあ、大変だソーサリアン！ジョセフィーヌを探し出せ！

5. アマゾンの剣（つるぎ）

女だけの町、ウァネルハの使者がヘンタウァへやってきた。
王が行方不明になり、魔物が街を荒しているというのだ！さっそく、何人かのソーサリアンがウァネルハへと旅立って行った

ゲーム性アップ&エンターティメント　羨望のユーティリティ発表。

SOCERIAN SYSTEM UTILITY Vol.1

『ソーサリアン』ユーティリティー・ディスク

「ユーティリティ・ディスク」はソーサリアン・システムの自由度を高め現ユーザーのニーズに答えたり、ファルコム「ソーサリアン」のスタッフが直接ユーザーへメッセージを伝える一種のメディアです。同一機種・メディアの「ソーサリアン」が必要です。

1. 道具の売買

「ソーサリアン」は自分の買って装備している武器や防具が同じ物でキャラクターどうしの道具の交換が出来ませんでしたがここで武器や防具を売買出来ますので、キャラクター間の武器の交換や自分の持っている武器や防具を一時的に売って保存しておくことも可能です。

2. 魔法をかける

「ソーサリアン」で自分の思い通りの魔法をかけるには相当大変でしたがここでは120種類の魔法を一瞬でかけて貰えます。

3. 名前の変更

「ソーサリアン」はキャラクターに名前をつけるときにアルファベットしか使えなかったのですが、ひらがなやカタカナの名前が付けられ　持っているアイテムの名前もかえられます。

4. ユーザーディスク・ツール

「ソーサリアン」に付属のユーザーディスクのバックアップや新しいユーザーディスクを作ったり、別々のセーブデーターのキャラクターの入れ換えが可能です。

5. BGM

「ソーサリアン」では機種毎に収録されているBGMが若干異なります。ユーティリティー・ディスクには未収録のBGMの一部が収録されており聞くことが出来るようになっています。

6.「ソーサリアン」クイズ

「ソーサリアン」にちなんだクイズを当社の最年少スタッフ「タッチャン」の司会でお楽しみ下さい。

7. お便りコーナー

「ソーサリアン」のユーザー・アンケート葉書をお送り頂いた方のご意見やスタッフのメッセージを当社の若きシナリオライター「いがちゃん」のDJでお送り致します。

8. ドラゴンと戦う

「ソーサリアン」では一定の条件を満たすとドラゴン軍団と戦えるモードが出現しますが、無条件でドラゴンと戦えるようになっています。(弱いと負けますが。)

9.「ミニミニ・ソーサリアン」

「ソーサリアン」の15本のシナリオを題材にしたゲームです。「ソーサリアン」のキャラクターが参加する形式で4人まで一緒に遊べます。最初にゴールしたキャラクターには賞品として経験値（EXP）が貰えます。「ソーサリアン」をプレイしたユーザーには大受けのゲームです。

「イースII」X-1も好評発売中！

イースIIは、イースの続きです。

Falcom　日本ファルコム株式会社
Personal Computer Software
〒190 東京都立川市柴崎町2-1-4 トミオービル

通信販売(送料無料)

●現金書留の場合
氏名・機種名・住所・氏名・電話番号を明記して、現金書留でお申し込みください。

●代金引換の場合
電話やFAXやハガキで、品名・機種名・住所・氏名・年齢・電話番号を明記して、お申し込みください。商品お届け時に商品代金をお支払いください。

TEL 0425(27)6501　FAX 0425(28)2714

7月31日まで
31日は棚卸しの為、6時閉店となります。

半期に一度！
夏よりHOTなツクモの夏！

ツクモはEXE Shop中のEXE
「スーパーXショップ」です。

1等10万円！
秋葉原電気まつり
8/7まで開催
賞金総額
7000万円
★5000円お買い上げの方に抽選券進呈。(東京のみ)

「ツクモX68000クラブ」会員募集

スペシャル会員
- 資格：当社にて本体ご購入の方
- 会費：1年間無料

レギュラー会員
- 資格：左記以外の方
- 会費：年間3,500円

TSUKUMO X68000 CLUB MEMBERS CARD 有効期限 会員番号

■うれしい特典たち■
- ホビー、ビジネスソフトの割引。
- シャープ製品(ソフト&ハード)の割引。
- 各種イベント、セミナーなどの優待及び割引。
- 会員証(テレホンカード)の発行。

そして、情報誌「X68000つ～しん」の配布。
その他数々の特典がわんさか、わんさか。

詳しいお問い合わせ、入会希望の方は ☎03-253-4199 (7号店・荒井へどうぞ)
※入会の手続きは、原則的に7号店店頭にて受付となりますのでご承了ください。

X68000 ACEシリーズ好評発売中！

CZ-611C	20MBハードディスク内蔵	定価¥399,800➡月々¥12,000×36回払など
CZ-601C	標準タイプ	定価¥297,000➡月々¥ 9,700×36回払など
CZ-601D	15型カラーディスプレイテレビ(0.39mmピッチ)	定価¥119,800
CZ-611D	15型カラーディスプレイテレビ(0.31mmピッチ)	定価¥145,000
CZ-6ST1	チルト台	定価¥ 5,800

X68000周辺機器もツクモで！

CZ-6VT1	カラーイメージユニット	定価¥ 69,800
CZ-8NS1	A4サイズフルカラーイメージスキャナ	定価¥188,000
CZ-6BN1	スキャナ用パラレルボード	定価¥ 29,800
CZ-6BE1	増設1MB RAMボード(CZ-600C専用)	定価¥ 35,000
CZ-6BE1A	増設1MB RAMボード(ACEシリーズ専用)	定価¥ 38,000
CZ-6BP1	数値演算プロセッサ	定価¥ 79,800

特別につくってもらいました。
CZ-52F(E)
X1Fモデル20又はX1ターボモデル20の方へ——5インチ内蔵型ドライブ
限定特価¥21,800

X68000用ソフトウェア

Kamikaze(神風)	統合型スプレッドシート	特価¥57,800
Z's STAFF PRO 68K	グラフィックツール	特価¥49,500
EW	日本語ワープロ	特価¥32,500
SAMPLING PRO 68K	AD PCM活用ソフト	標準価格¥17,800
COMMUNICATION PRO 68K	通信ソフト	標準価格¥19,800
スウィング PRO 68K	リレーショナルデータベース	特価¥49,300
C-TRACE 68	レイトレーシングソフト	特価¥59,800
CONCERTO-X68K	MS-DOSエミュレータソフト	特価¥99,000

その他、ビジネスソフト・ホビーソフトも多数発売中ですのでお気軽にお訪ねください。

X68000用ハードディスク

●アイテック IT-H540HX	40MBタイプ	28ms	特価¥133,000
●ウインテク HD-202	20MBタイプ	85ms	特価¥ 72,000

ポケコンコーナーへどうぞ
シャープ PC-E200 定価¥22,000
Z-80CPU、RAM容量32KB
ツクモ特価¥17,800
PC-E500 定価¥28,800
32KB標準装備
ツクモ特価¥24,800

本当にこのお値段でいいんです。

パソコンテレビ X1G モデル30
- CZ-822CB ¥118,000
- CZ-820DB ¥109,800
- ディスケット(10枚)+ゲームパック サービス

合計定価¥197,800
ツクモ特価 ¥99,800
★月々¥8,900×12回払いなど分割もOK!!

X1 turbo Z セット
- CZ-880CB ¥218,000
- CZ-880DBK ¥109,800
- ディスケット(10枚)+ゲームパック サービス

合計定価¥327,800
ツクモ特価 ¥188,000
★月々¥8,900×24回払いなど分割もOK!!

X1G モデル10
●CZ-820C ¥69,800
6種類のソフトがパックされた「横綱」がついています。
ツクモびっくり特価 ¥19,800

信頼と安心から生まれたツクモオリジナル

5インチ2Dドライブ

- TS-FDMKIIにケーブル及び特製I/Fをセットしたもので、これだけでディスクシステムが使用できます。
- 1ドライブはCZ-503F、2ドライブはCZ-502F相当品です。

1ドライブ特価¥32,800
2ドライブ特価¥49,800

5インチ2HDドライブ TS-FDD MKII X1(ターボモデル10を除く)
X1ターボ用2HD/2DD自動切替
1ドライブ特価¥38,800
2ドライブ特価¥59,800

X1ターボ/MZ-2500用
マウス
TS-MX1
特価¥4,800
■便利なマウスパッドあります。¥1,280より

モデム

関西電機	KDM-3012L	300/1200ボー	ツクモ特価¥16,000
オムロン	MD-2400B	300/1200/2400ボー	ツクモ特価 ¥42,800

プリンター

CZ-8PC2	カラー漢字熱転写プリンタ	ツクモ特価販売中
CZ-8PK5	24ピン漢字ドットプリンタ(10インチ)	ツクモ特価¥69,800
CZ-8PK6	24ピン漢字ドットプリンタ(15インチ)	ツクモ特価¥89,800

全国代金引き換え配達
お申し込みは☎03-251-9911へお電話1本！
商品到着の際、玄関でお会計ができます。配達日の指定もできます。

ツクモらくらくクレジット
月々¥3,000以上の均等払いも頭金なし。
ボーナス月加算払いを併用して欲しいもの先取り！

現金書留なら
〒101-91 東京都千代田区神田郵便局私書箱135号
九十九電機(株)通信販売部

銀行振込なら
事前に☎でお届け先をご連絡下さい。
富士銀行 神田支店(普)No. 894047

PRO STAFF ツクモ
九十九電機(株) 〒101-91 東京都千代田区神田郵便局私書箱135号

通信販売部 ☎03-251-9911
ツクモ5号店 ☎03-251-0531
ツクモ7号店 ☎03-253-4199
ツクモニューセンター店 ☎03-251-0987

THE SOFTOUCH

SOFTWARE INFORMATION

スペースハリアー
M.U.L.E.
ラスト・ハルマゲドン
エグザイル・破戒の偶像
ARCUS
ヘルツォーク
ザ・リターン・オブ・イシター
R-TYPE
アドベンチャーゲームインタープリター電脳作家
Toys & Tools Human68k

完成も間近となったX68000のドラゴン・スピリット，R-TYPEとイシターです。そして最後はX1ユーザーお待ち兼ねのハイドライド3

話題のソフトウェア

今月のSPECIAL REVIEW，もう読んでくれたかな。いま注目のイースIIがちゃんと載ってたでしょ。6月24日発売の最新ソフト情報をどうしてもこの8月号でお伝えしたくて，一生懸命ガンバッたんだよ。また7月にはT&Eから**ハイドライド3**が発売される予定だし，RPGファンはとっても忙しい夏を迎えそう。でも，これらのRPGを見ていると，それまで本格派RPGの世界を目指していた基本路線から，最近では新たに**冒険活劇RPG**と呼んでもいいようなひとつのジャンルが完成されつつあるような気がするね。

そのジャンル分けはともかくとして，欲張りなゲームファンのひとりとしては，いまのうちからこのイースやソーサリアン，そしてハイドライドの次に来るものにぜひ期待しておきたいものだね。

さて，皆さんお待ちかねの今月のX68000最新ゲーム情報からいってみましょう。まずはSPSから**ザ・リターン・オブ・イシター**が，そしてテクノソフトからは**サンダーフォースII**，さらにはリバーヒルソフトの**琥珀色の遺言**とJDSの**名監督**，そして先月A列車IIの発売を伝えたばかりのアートディンクからは**アークティック**が発売される予定だって。上の写真を見てもわかるようにやや発売の遅れているドラゴン・スピリットや8月発売予定のR-TYPEも順調に仕上がっているようだし，都内某所では**アフターバーナー**プロジェクトがスタートしたらしいし，コリャ，ますます楽しくなりそうだ。ところで，先月号で7月発売予定だとお伝えした熱血高校ドッジボール部は，ちょっと遅れて秋頃になるみたい。もう少し待っててね。

読者が選ぶ今月のゲームベスト10

発売されたとたんに上位に入ったのがソーサリアン。やっぱりね。3位以下を大きく引き離して源平討魔伝とトップ争いを繰り広げています。パソコンゲームで初めて「剣と魔法」の世界を知ったという人が，以来「アーサー王」を座右に置いてる，なぁんて言ってましたっけ。ゲーム大好きでよかったね。

イースIIも見逃せません。次はきっと上位に入ってくるでしょう。うん，今度どなたか順位予想をやってみませんか？　宛先はOh!X編集室ゲームベスト10係です。

1.源平討魔伝
2.ソーサリアン
3.SUPER大戦略
4.三国志
5.スペースハリアー
6.イースII
7.イース
8.スーパーレイドック
9.ぎゅわんぶらあ自己中心派2
10.上海

さて，これまでX68000関係のソフト情報は，ゲームはこの「SOFTWARE INFORMATION」で，ツール類は「X68000あなたの知らない世界」でと2カ所で紹介してたわけだけど，ずいぶんと発売されるソフトの数が多くなってきそうな気配なので，来月からは一本化して，このTHE SOFTOUCHのなかに新しく**X68000ソフト情報コーナー**を新設したいと思っているわけ。誰かそのコーナーに付けるいいネーミングがあったらハガキに書いて送ってくださいな。採用された方には1988年型最新モデルのOh! Xロゴ入りオリジナルシャープペンをドーンと豪華に1本差し上げちゃいましょう。いやー，本当に太っ腹のOh! X編集室ですこと。そいじゃ，よろしくね。

新作ソフト情報

☆…7月2日現在発売中　★…近日発売予定

★スペースハリアー

スペハリと聞けば，X68000版のことだと思っていたX1ユーザーのためにX1版がついに発売されることになった。ゲームの内容なんてもう説明するまでもないが，意外とそのシナリオについては知らない場合が多いので，あえてここで簡単に紹介しておくと，超自然現象と凶悪な魔生物に侵略されたドラゴンランドを救うべく，超能力戦士ハリアーはオートロック式ランチャーを手に単身戦いを挑むのだった……，というもの。さあ，あとはドムやアイダなどお馴染みの敵キャラ相手に，3Dシューティングアクションを存分に楽しもう。

X1/X1turbo用　5"2D版2枚組　7,800円
（2ドライブ専用）
電波新聞社　☎03(445)6111

スペースハリアー

M.U.L.E.

★M.U.L.E.

シミュレーションとひと言で表現するには，あまりに毛色の違ったゲーム「M.U.L.E.」がX1に登場だ。このゲーム，簡単に説明すると遠い惑星で資源調査と開拓を進めていくわけだが，その間，自分の土地を確保したり食料を生産し，オークション会場にも足を運び，そこで儲ければ多目的ロボット・ミュールを買付け，自分の土地に設置していくというもの。一見地味だが，4人プレイで遊べば，プレイヤーの個性が出てしまって，とっても熱くプレイできそう。

X1/X1turbo用　5"2D版　7,800円
ビー・ピー・エス　☎045(931)5815

★ラスト・ハルマゲドン

このRPGの舞台は，いまから何万年も先の未来。人類は滅亡し，地球を支配するものはいなかった。そこに現れたのが，かつて人類によって滅ぼされたはずのモンスターたち。地球を我々のものに，と奮い立つ彼らの未来は順風満帆のように思えた。が，そんなとき彼らの前に立ちはだかる生物が。人間？　いや，そんなはずはない。彼らは滅びたはずだ。その生物はモンスターたちと同じく，地球を支配しようと宇宙からやってきたエイリアンたちであった。そしていま，地球をめぐってモンスターとエイリアンの戦いが始まる。このラスト・ハルマゲドンの主役は人類ではない。そう，モンスターたちなのだ。

X1/X1turbo用　5"2D版5枚組　7,800円
（2ドライブ専用）
ブレイングレイ　☎03(264)3534

★エグザイル・破戒の偶像

このエグザイルは，アクションシーンとRPGシーンの2つで構成され，アクションシーンは源平討魔伝のデカキャラモード，RPGシーンはドラクエタイプの画面構成となっている。そしてこのゲームは12世紀という時代設定からスタートして，9つあるシナリオを，1つひとつ解いていくうちに，このゲームの最終目的を知ることとなり，舞台も20世紀へと移っていくようだ。アラビア半島を中心に展開される古代の神秘な世界。はたまた現代の中東情勢がゲームに取り込まれていて楽しめそう。また，発売日より2週間以内にこのソフトに付いている応募用紙を送ると，もれなく日本テレネットのオリジナルミュージックCDが当たるオマケも付いている。

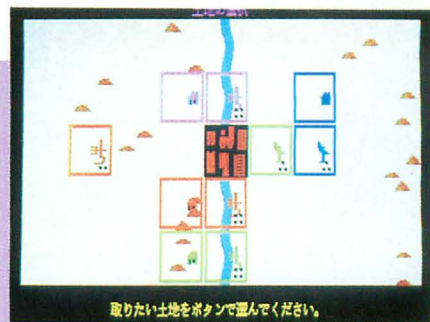
X1turbo用　5"2D版3枚組　8,800円
（2ドライブ専用）
日本テレネット　☎03(268)1159

★ARCUS

「RPGには様々なタイプがあるわけだが，すべてのRPGにおいて共通点がある。それは経験値である。（中略）このことについては最近，経験値稼ぎという言葉が生まれた。シナリオやグラフィックが変わっただけで基本的なシステムは変わっていなかった。そこでわがウルフチームは，考えたのである。これまでのパワープレイを中心としたRPGを第一世代のRPGと例えるなら，第二世代のRPGはテーブルトークタイプと考えたわけである。レベル，経験値はなく，会話を中心とし，キャラの成長はデジタル的ではなくアナログ的にする。いま，RPGが新しくなる」（「RPGにおけるARCUS概括論」より一部抜粋）。ここからもわかるように，あの，ウルフチームの意欲作がX1に初登場だ。88版のものにかなりの変更が加えられる予定らしく，ウルフチームの実力に期待したい。

X1/X1turbo用　5"2D版5枚組　9,800円
ウルフチーム　☎03(269)8650

☆ヘルツォーク

2人同時プレイもできるリアルタイム・シミュレーションゲームが登場だ。あなたの使命は機動歩兵と破壊工作兵器を使って，敵本拠地を攻めるべく前線基地で戦い抜くこと。2人同時プレイの場合には，同じ画面のなかを動き回るのではなく，画面を縦に2分割し，プレイヤー1とプレイヤー2が別々のスクロールをしてしまったりと，いままでのシミュレーションとは毛色の違うものに仕上がっている。このあとのGAME REVIEWでも紹介しているので，そちらのほうも参考に。

X1/X1turbo用　5"2D版　6,800円
（要G-RAM，turbo以外はジョイスティック専用）
テクノソフト　☎0956(33)5555

★ザ・リターン・オブ・イシター

このゲームはあのドルアーガの塔の続編で，ギルがドルアーガを倒し，その魔力を封じ込め，恋人のカイとともにこの塔から脱出するところから始まる。しかし，ドルアーガの魔力が解けたためにその魔力によって修復されていた塔は廃墟と化し，出口への道も迷路となってしまった。さらにドルアーガの呪縛が解けてしまったため，モンスターたちは以前にもまして狂暴になっていた。果たして，ギルとカイは世界を救うことができるか，それはあなたの腕次第。

X68000用　5"2HD版　7,800円
エス・ピー・エス　☎0245(45)5777

★R-TYPE

昨年，ゲームセンターで好評を博したR-TYPEが各機種に移植されているのは皆さんご存じのとおり。首を長くして待っていた方，お待たせしました。いよいよX68000にも登場だ。次々に現れるグロテスクな敵キャラを相手に，あなたは単身R-9に乗り込み，戦わなければならない。果たして，バイド帝国の支配者バイドを倒すことができるだろうか。どこまでアーケード版の迫力に迫れるか。

X68000用　5"2HD版　7,800円
アイレム販売　☎06(535)4480

☆アドベンチャーゲームインタープリター
電脳作家（サイバーライター）

ついに，X68000ユーザー念願のアドベンチャーゲーム用のコンストラクションキットが登場した。Human68kのコマンドとして扱うことができ，このツール専用のグラフィックツールも付いている。なお，このアドベンチャーツールを使ったアドベンチャーゲームシナリオコンテストも予定されており，入選作品の一部は，商品化が行われるほか，それ以外の作品は各応募作品ごとに得点順位，偏差値，審査員のコメント付きで返送される。

X68000用　5"2HD版2枚組　4,980円
日コン連企画　☎06(644)6901

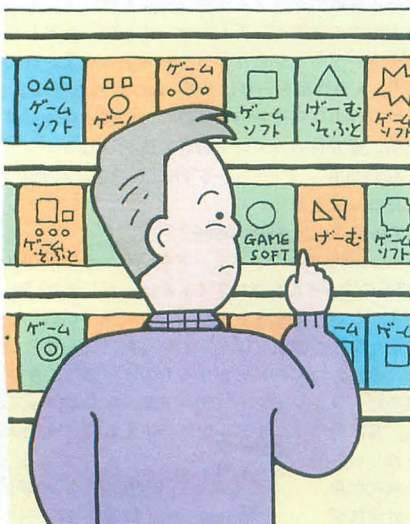
☆Toys&Tools Human68k

Human68kのコマンドモードを使ううえでACOPY（複数のファイルを連結する）といった，あれば便利！　というようなものから，QIX（見ているだけで心がなごむ環境ソフト）といった，こんなのなんの役にたつんだと思わせるものまで幅広く各種集めたトランジェントコマンド集。なんといっても，基本的にマニュアルレスで使える親切設計がうれしい。

X68000用　5"2HD版　6,800円
計測技研　☎0286(22)9811

THE SOFTOUCH

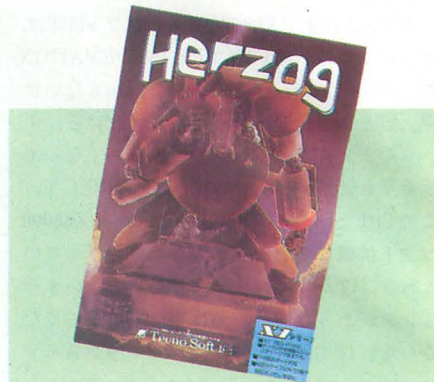
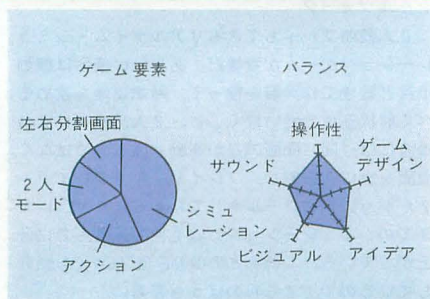
GAME REVIEW

今月は，シューティングアクション「ヘルツォーク」と，正統派麻雀ソフト「プロフェッショナル麻雀悟空」。そしてRPGの「ハーレッシュ」の3本です。うまくすればX68000のドラスピが間に合うかと思っていたけど，今月もハズレ。来月，期待していてね。

ヘルツォーク

2プレイヤーの場合だと，左右に2分割された画面が上下別々にスクロールするシューティングゲーム。熱くなること請け合い。

▶パッケージを見て，ひと昔前のスクロールシューティングかと一瞬不安を感じる。デモを見て，さすが技術のテクノソフト，いい処理してるなあと感心する。そしてゲームをプレイしてみると，やっぱりヘルツォークにしてよかった！　となる。実際，このパッケージは地味だし，なんとなくつまらなそうな表情のゲームである。ところがぎっちょん，真実はさにあらずである。プレーするとわかるが，このゲームの戦略的要素は大戦略を，2分割スクロールはボールブレイザーを連想させるほどの出来栄えなのだ。前評判はないに等しいこのテのゲームでは，昔の悪夢のような操作性が思い出されそうだが，その心配もない。リアルタイムの要素とストラテジックな要素が非常にうまく調和していて，妙にハマったりしてしまうソフトだ。とにかくシューティングというよりシミュレーションの色合いが強いので，その筋がお好きな方は，ぜひ2人用モードで実際にプレイしてみてほしい。

熱中度▶▶▶▶▷▷▷　(A.N.)

▶まずはなにも見ないでゲームをやってみる。1分たった。しぇー，なんじゃこれは。訳がわからん。今度はマニュアル片手にやってみる。なになに，破壊工作兵器を作って，敵の前線基地を破壊すればいいのだと。こりゃ単純なゲームだこと。それならば，

ジャンジャン兵器を作ってやろうじゃないの。それっ，戦車だ，核ミサイルだ！

おっ，こいつは楽しいや，作った戦車やら歩兵やらが，前線基地に向かってトコトコと勝手に動いてくれるではないか。おまけに，敵と衝突すれば戦闘だってしてくれる。それをコーヒーでも飲みながら，ノンビリ眺めてりゃいいのかと思うと，世の中そんなに甘くはなかった。自分はといえばモビルスーツを操縦して，兵器の輸送やら，敵の攻撃やらで，コセコセと画面を動き回って大忙し。このゲーム，人間と戦うこともできるので，コンピュータ相手にムッツリするよりは，友達相手にギャーギャー騒ぎながら遊ぶほうが健康的でしょう。

熱中度▶▶▶▶▷▷▷　(H.K.)

X1/X1turbo用　5"2D版　6,800円
(要G-RAM，turbo以外はジョイスティック専用)
テクノソフト　☎0956(33)5555

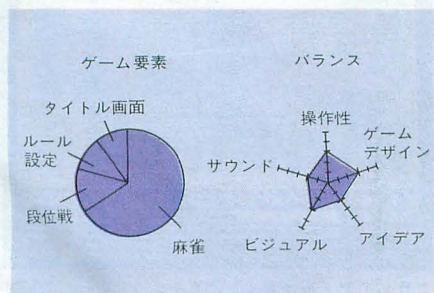
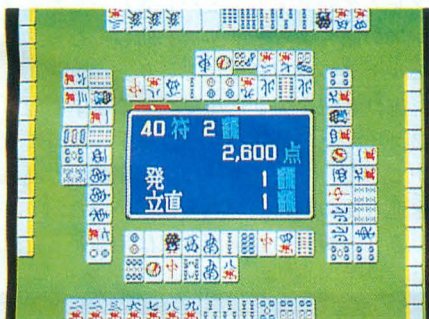
プロフェッショナル麻雀悟空

24人の対戦相手のなかから雀士を選び，段位戦，勝抜戦，段位の3つのモードで対戦できる正統派4人打ち麻雀ゲームです。

▶うーん，凄い。なにが凄いって，この麻雀悟空，4人打ち麻雀で結構強いんですよー(例によって私が弱いだけかもしれないが，そんなことは気にしない)。で，麻雀悟空という名前からもわかるように，麻雀に出てくるキャラが悟空だったり，八戒だったりするわけで，1人ひとりのキャラの癖が(そうそう，1回1回対戦する相手が違うんですよ)平均和了点25000点とか，立直率.250とか，数字で見られるんですよー，うーん恐ろしい。でもこれだけキャラをリアル(っていうのはなんか変な感じだけど)にするんだったら，なんで，上がったときの「ちゃい」とか「これで勝ったと思うなよーっ！」とか，リアクションを付けてくれなかったんだろう？

はっきり言って，3人のゴルゴ13にかこまれて麻雀やってるみたいでいまいち，落ち着けないんだよねー。黙々と麻雀やる人にはいいのかもしれないけど……。あと0.5歩で完璧なのになー。

熱中度▶▶▶▶▷▷▷　(で)

▶右手人差し指を←に，中指を→に，薬指を↓に添え，歴戦の雀士を前に，牛魔王，三蔵，敵は多く，ルールはこちらで決められるが，あまりローカルなのも気が引け，三連刻くらいはいれてもらうがカンウラはなく，ぼんやりしていると鳴き損ね，うっかり捨てると振り込み，時折数秒の長考に入るかと思えばおもむろにリーチをかけられたりして，それでも早朝，まだ明けやらぬ空の下，これから1日が始まろうというすっきりした頭で挑めば，慎重に慎重を重ね，無理をせず，振り込まずして勝てぬこともなしとはいえ，半荘も3度目を数えると朝刊も配達され，多少計算機相手に「このやろう」などと口走り始める頃には，あれよあれよと裸にされ，それでもコーヒーの持つカフェインの力で持ち直し，前半の貯金にものをいわせ，なんとか“悟浄”は初段へと昇進した。

実戦並みの緊張感と頭脳を要求される，久々のまっとうな麻雀遊戯であった。

熱中度▶▶▶▶▶▶▷　(K)

X1/X1turbo用　5" 2D版　6,800円
シャノアール　☎03(702)0598

ハーレッシュ

過去に封印されたはずの影の神が，突如復活した。そこで光の戦士が失われた光を求めて立ち上がるという，RPGなのです。

▶このハーレッシュは，いわゆる普通のアクションRPGとなっております。町の住民から情報を集め敵と戦って経験値やお金を稼ぐ。そいでもって，剣や防具を買ったり修理をしたり，ときにはデカキャラも登場したりするアレです。スクロールは波打つし，画面消去は遅いし，BGMは変化に乏しいし，マニュアルは不親切だし，Oh！Xの編集者はオタクしてるようだし（失礼，これは関係ない）と，欠点は多いのですが，ところがどっこい。プレイを続けていると簡単に解けていく謎，そして次々と変化しながら広がっていくマップなど，なかなか心地よいのです。過去に，あのザナドゥを5時間で飽きたこの私が，18時間もの間，すっかり夢中になってしまいました。

いま話題のイースIIやソーサリアンは，周知のとおり，すっかりturbo専用になっていますので，RPGファンのX1ユーザーの方は，ぜひこちらを召し上がってみてはいかがでしょうか。

熱中度▶▶▶▶▷▷▷　(お)

▶リアルタイムのアクションRPGである。最近は，戦闘が起こるといちいち戦闘画面とかに切り換わったりするのが多いが，このゲームは，敵のいる方向に押しつけると，勝手にボカスカと殴り合ってくれるという，実に安直な方式でなかなかよろしい。傷直しをしなくてもどんどん傷が直っていくのもいい。

グラフィックはなかなかきれいである。人に会って話を聞くとき，クローズアップ

が出てくるのはなかなかよい。また，宮殿にはいかにもという感じのBGMが流れ，衛兵が行進をしていたりしてなかなか雰囲気がよい。お姫様に会いに行くところもそれっぽくてかっこいい。

難を付ければ，ファミコンなんかのゲームに比べて，詰めが甘いというか，いまいちのかったるさがあるのが問題といえる。しかし，パソコンRPGにありがちな，ごてごてとしたところを廃した，バランスのいいゲームである。

熱中度▶▶▶▶▶▷▷　(M.Y.)

X1/X1turbo用　5" 2D版3枚組　7,800円
ザイン・ソフト　☎0794(31)7453

私はもっと自由に空を飛びたいのだっ！

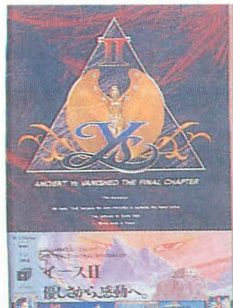
ちわーす，オタッキーの(で)でーす。違うってば。ところでこの私は，X68000というと「ザ・コックピット」でばかり遊んでいます(もちろんX1ではまじゃべんちゃーだけど)。このコックピットは，これはこれでとっても満足してるんだけど，やっぱりセスナ機や戦闘機みたいにキリモミ飛行はできないし，急上昇なんかもできないんですよねー。それに私は夕焼けに染まる空港を飛び立ちたいし，昼の景色も見てみたい。ほかにもマッハ2でほかの飛行機とも競争したい。したいっっ！　そう，もっと私は自由に空を飛びたいのだぁ。最近，この私が遊べたフライトシミュレーションって，どうもコックピットのほかには，ずいぶん前の投稿のMZ-1500用「SKYHOLYDAY」しかないじゃないですか。これではいけないっ！　ねー，誰かX68000にもっと凄いフライトシミュレーションを作ってよ。

（すっかり他力本願の(で)）

●イースII(第1話)

少年アドルの新たなる旅立ち

Kamon Masato
華門 真人

6冊のイースの本に秘められた新たなる謎に立ち向かうため，少年アドルは壮大な冒険の旅へと旅立つ。音楽，デザインなどは言うに及ばず，なかでも今回のイースIIでは，完成されたストーリーにぜひ注目していただきたい。

X1turbo用　5"2D版4枚組　7,800円
(Model 10では要CZ-8BGR2，CZ-8BF1 2ドライブ専用)
日本ファルコム　☎0425(27)6501

第1章 ランスの村

「大丈夫？」
かわいいその声で，ようやく俺は意識を取り戻した。
「ここは？」
どうやら，またも見知らぬ土地に来てしまったようだ。故郷が妙に懐かしい。

しかし，そんな感傷に浸っていられたのもバノアの家を訪れるまでのことだった。そう，俺は助けてくれたリリアなる娘に連れられランスの村へ向かったのだ。ランスの村では，まずリリアの母バノアを訪ねた。

そこで俺は悲しい事実を聞かされた。あのリリアは実は重い病いに侵されているという。なんとかしてあげなければ。そう思うと，俺は再び胸のうちに熱いものが燃えさかるのを感じた。そう，俺は行かなければならない。リリアのために。

第2章 廃虚

バノアの話によると，医者のフレアならばリリアのための薬を作れるという。とりあえずフレアを探そう。そう思ってバノアの家を出て村を歩きまわっていると，フレアの弟なる人物に出会った。

彼の話によると，フレアはいまは廃虚と化している廃坑のなかで生き埋めになっているらしい。俺の新たな目的は決まった。手元にはバノアにもらった300Gがある。俺はそれで剣と楯，そして薬草を買い求めた。村人たちの話を聞くだけ聞いたらいよいよ廃坑へと出発だ。

廃虚のなかに入ると，さっそくモンスターどものお出迎えだ。久々の戦いに最初は戸惑う俺だったが，カンさえ取り戻せばこっちのもの。俺は次々とモンスターどもをなぎ倒し，体力の充実に努めた。

廃虚のなかで俺は奇妙な老人と出会った。その老人は俺にモンスターの守っている宝箱のなかの物を持って来てくれるように頼むのだ。よしきた，とばかりにそのモンスターに戦いを挑んだのはよかったが，相手が強すぎた。

やっとのことで逃げ出した俺は，さらに体力を充実させ，装備もふやして再び戦いを挑んだ。1回相手を切りつけては離れるというヒット・アンド・アウェイ戦法をとった俺の前に，モンスターはなすすべもなく消滅した。

こうして俺は神界の杖を手にした。先ほどの奇妙な老人の教えに従い，女神像の前に立った俺に奇妙な変化が起こった。そう俺の体に魔力が備わったのだ。

俺はさらに廃虚の探索を続け，ロダの実などを手にした。そして十分に力をつけた俺は，いよいよ目的地である廃坑へと向かうことを決意した。

第3章 廃坑

俺が村に戻ると，村長に出会った。村長は廃坑に入ることを快く許してくれた。こうして俺は廃坑へと向かった。

廃坑に入ってしばらく行くと，奇妙な部屋を見つけた。静まり返ったその部屋の奥には神官の像が立っていた。その像の前に立ったとき，たたずんでいる俺を驚かせるようなことが起きた。イースの本が1冊，音もなくその像へと吸い込まれていったのだ。そしてあの声が聞こえてきた。

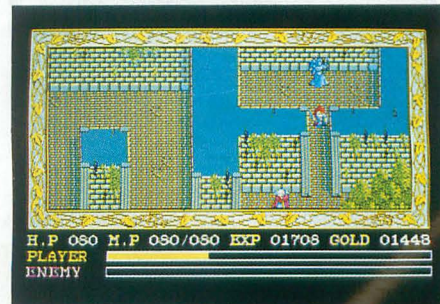
声の主は行方不明と伝えられていた神官のひとりであった。彼の口からことの次第が語られた。

俺は廃坑をさらに奥へと進んだ。そしてイースの本を次々と神官に返していった。こうして5冊のイースの本を返した俺は，ランスの村で起きていた不可思議な出来事の全容を理解した。そしていまや，この天上国イースも魔物の危機にさらされているのだ。そこで地上を救った俺がイースを救うべく呼び寄せられたのだ。

もちろん俺は本来の使命も忘れてはいなかった。俺はフレアを助け出さなければならないのだ。強力な敵になんどかくじけそうになった俺だが，その度に神官の部屋で休息をとり，冒険を続けた。そしてついにそれらしき場所を見つけることができた。

俺は渾身の力を込めてマトックを振り下ろした。そして壁が崩れ，そのなかにいたフレアを発見することができた。しかしフレアの話によるとリリアを助けるためには，セルセタの花が必要だという。

俺は廃坑を突き進んだ。その途中で俺はファイヤーの魔法なるものを手にすることができた。ファイヤーの魔法を装備し廃坑を進む俺の前に奴が立ちはだかった。

そう，親玉の登場だ。奴は手強かった。

こうしてアドルは魔法が使えるようになったのです

口から弾を吐き出すのだ。しかし俺はファイヤーの魔法で果敢に戦った。狙うは奴の口だ。そして危ういところで俺は奴を倒すことができた。

こうして俺は廃坑の第2層へと突入した。第2層ともなるとモンスターどもはさすがに手強い。それでも俺はなんとかセルセタの花とライトの魔法を手に入れることができた。廃坑はこれで終わりであるかのように思えた。が，しかし，ふと思いたってライトの魔法を使ってみると，見えなかった2つのゲートが輝きを放ったのだ。俺はそこで邪悪な鈴と鉄鉱石を見つけ出すことができた。

こうして，廃坑の探索は終わった。そう確信した俺はウイングを使ってランスの村へと戻ることにした。

第4章 地下室

村に戻った俺はさっそくフレアを訪ね，リリアの薬を調合してもらった。これでリリアは助かる。俺は安堵した。

しかし俺はそうそう休んでいるわけにもいかなかった。俺にはイースを救うという重大な使命があるのだ。どこに行けばよいのだろう。廃坑にひとつ開かない扉があった以外，思いつく場所がない。

そんな俺の耳に，ジラの家の地下室に魔物の気配がする，という情報が飛び込んできた。俺はさっそくジラの家へと向かった。ジラの地下室に入った俺は，なにか怪しい気配を感じた。

「なにかが，いる！」

俺はそう確信した。

「来るなら来い！」

邪悪な鈴の音が地下室に響きわたると同時に壁がすさまじい勢いで崩れ落ち，そしてモンスターどもが現れた。

モンスターをファイヤーの魔法でけ散らし，壁の向こうに入った俺は，すぐに神官ファクトを見つけることができた。

ついに6冊目のイースの本が音もなく神官の像へと吸い込まれていった。そして俺は新たな使命を与えられた。神殿に行って2人の女神を助けねばならない。イースの平和のために。

導きの巻物を手に，俺は例の「開かずの扉」へと向かった。扉をくぐった俺の目の前に，極寒の世界が広がっていた。本当の冒険はいまここから始まったのだ。

やっぱりイースっていーす

さて，イースIIのレビューはいかがでしたか。なになに，まだ全然終わっていないじゃないかって。それは当然ですよ。こんなに大きなストーリーを1回で紹介するなんてとてもじゃないけど無理ですからね。というわけで冒険物語イースはさらに来月へと続くのです。

しかし，このストーリー，本当に長いですよ。なにしろマップの大きさは前作の3～4倍だそうですからね。さらにここから話は地下室から氷の国，火の国を経て，ようやく神殿へと行き着くのです。

それにしてもイースの操作性の良さは今回も健在です。スクロールは極めて滑らかですし，キーのレスポンスは抜群です。ただ，敵のキャラが非常に多くなると一時的にスピードが落ちるという現象がジラの家の地下室で見られたのですが，ほかのところでは一切ありませんし，それもほとんど支障のないような程度なので許しましょう。

こうした快適な操作性と相まって，イースの魅力となっているのは，その綿密なまでに計算し完成されているバックボーンとなるストーリーです。プレイヤーはまさに，ゲームの進行とともに物語の主人公になりきることができるわけです。ですから，RPGといってもつまらないだけの経験値稼ぎや，金集めをする必要がないのです(もちろんやろうと思えばいくらでもやれますが)。

すなわち，物語を進めようと努力すれば経験値にしろ金，レベルにしろ自然にあとから付いてくるのです。イースではアイテムでさえ敵に勝つためのものではなく，物語を進行させるためのものに過ぎないのです。

最近はこういった種類のRPGも増えてはきましたが（これもすべてイースのヒットによるものであると私は思っている)，やはり本家本元だけあってその完成度の高さは目を見張るものがあります。そもそもゲームっていうのは，楽しむためにあるものですからね。だからこのイースの，とことん最後までプレイヤーを楽しませようという姿勢には感心させられます。

イースといえばBGMの魅力も忘れられませんが，今回はFM音源を使い，曲もたいへん豊富になっています。神官の部屋では落ち着いた感じの，村では明るい感じの，廃坑では暗い感じのといった曲の使い分けもしっかりしており，雰囲気を盛り上げてくれます。

それに付け加えるならば，X1版ではセーブが7カ所もできますから（これはX1版だけですよ！)，かなり解きやすくなっています（でも，やさしいだけというわけでは決してありません)。致命的なハマリこそないといっても，たくさんセーブできるにこしたことはないですからね。

とまあ，操作性，画面，スケール，そしてストーリー，BGMとどれもが前作のよさを引き継ぎながらも，このイースIIはより向上しているのです。これはやらないわけにはいきませんよね。というわけで「やっぱりイースはいーす」，など使い古されたシャレを残しつつ，広大なマップの征覇を目指してまた来月。

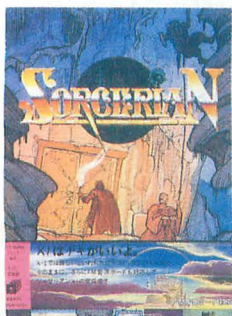
アドルは廃坑のなかで美しい花を見つけた(写真左)。これをフレアに渡せばリリアをきっと助けることができる（写真左下)。しかし，廃坑にある最後の扉が開かれたとき，アドルの冒険の旅は，氷の世界から最後の神殿までと，まったく新しい世界のなかで，さらに続くのです

THE SOFTOUCH

●ソーサリアン(その2)

クイーンマリー号
徹底解剖の巻

Nishikawa Zenji
西川 善司

ソーサリアンはRPGとしてもAVGとしても楽しめる要素をたくさん持っています。そこで今月は，スグレもののシナリオとして西川善司氏ご推薦の「呪われたクイーンマリー号」にスポットを当てて，その徹底解剖をお届けしましょう。

X1turbo用 5"2D版5枚組 9,800円
(model10では要CZ-8BGR2，CZ-8BF1)
日本ファルコム ☎0425(27)6501

こんにちは，西川善司です。

まず初めに，先月のレビューの訂正をしておきます。24ページの「暗き沼の魔法使い」のシナリオ紹介で，「バシバシ2段ジャンプを使って〜」というところで，その文の前に「魔女の鍋はゾンビの出るところで」という文章が抜けているんです。これはミスプリントでした，ごめんなさい。

それと「音楽は11和音」と書いてしまいましたが，前回出没した吉田賢司君によれば「4MHzのZ80には11和音は重荷だよ。PSG3声はいつも使っているみたいだけどFM音源は5〜6声しか使っていないみたいだよ」だそうです。

シナリオ徹底研究ナノダ

さて，今月は私の大好きな「呪われたクイーンマリー号」を徹底レビューしてしまいましょう。このシナリオはRPGとしては珍しく，1隻の船が舞台であります。船内で起きた殺人事件の犯人を捜したりするイベントなんかもあって，「推理アドベンチャー」の要素をも持っています。少し練り直せば，これ単体で売っても恥ずかしくないような出来のシナリオで，難易度もかなりのものです。そして推理はアドベンチャーの大御所，J.B. ハレルヤ氏にご協力いただくことにしましょう。

それでは，はじまり，はじまり〜。

ACT1 セントエルモ

J.B.は，潮風にしばしの別れを告げ，事件現場の「クイーンマリー号」に乗り込んだ。暗い階段を上って船上に出ると，そこには杖をついた見るからに船長という人物がマストに寄りかかっていた。

「ハレルヤさんですね。わざわざ，来ていただいたのに申し訳ないのですが，この船に乗るのはやめたほうがいいと思います」

J.B.は，船長に事件について尋ねた。

「はい，以前ある島に立ち寄ってから，おかしなことばかり起こるんです。船員のデビッドが殺されたのもその直後でした。船員たちは船倉になにか居るらしいと口々に言っていますし，本当はこんな船に息子を乗せておきたくはないのですが，息子はペットのインコに逃げられてしまってからというもの，すっかりふさぎこんでしまいまして……」

J.B.は船長と別れ，船員の部屋へと向かった。

アンドリューという船員の部屋のドアを軽くノックしたあと，部屋に入った。

「誰だい，あんたは」

J.B.は事情を説明したあと，事件について聞いた。

「し，知らんそんなこと……」(これはマンハッタン・レクイエムでよくみられる台詞のひとつですね)

J.B.はなにか情報はないかと再び尋ねた。

「そういえば，ネズミがインコの籠を引いていったぜ」

J.B.はアンドリューの部屋を出てボブの部屋を訪ねた。

「デビッドは，船倉で殺されていた。最近，船倉がどうも騒がしい」

そこで机の引き出しになにかあるかと思って調べたが，鍵が掛かっていて開かなかった。仕方なくボブの部屋を出た。

J.B.は，もう1階下の船室を訪ねようと階段を下りた。この船は船倉も含めると3層構造になっているらしい。そして今度は殺されたデビッドの部屋に入った。きれいに片付けられていたが，なにか嫌な雰囲気をJ.B.は肌に感じながら部屋のなかを捜してみた。するとデビッドの日記が見つかった。さらに捜してみると机の引き出しに船倉の鍵を見つけた。

J.B.は，キリーの部屋を訪れた。

部屋のなかには赤いヒゲを生やした，年配の船乗りがいた。

「ハレルヤさん，船長の息子を元のセントエルモのような明るい子に戻してあげてください」

J.B.は，この「セントエルモ」という言葉に聞き覚えがあった。そう，確か昔，船乗りの友人から船のマストに夜になると青白い炎のようなものが見えるという話を聞いたことがある。J.B. は，船上に上がるとマストに登り，夜を待った。

ACT2 船長の息子

樽が沢山ある部屋を抜けるとひとつのドアがあった。この部屋は暗く，湿っぽかった。J.B.は部屋の奥で船にぶつかる波をじっと見ている少年に向かって話しかけた。

「おじさんは誰？ そう，パパに頼まれて

デビッドの部屋で日記と鍵を発見

来た探偵さんなの。でもパパは僕のことを嫌いみたいだよ。だって突然，わけも言わずに船を降りろなんて言うんだ」

傷ついた少年の心を癒す術を知らなかったJ.B.は，少年に別れを告げ，船倉のいちばん奥へと向かった。

ACT3 船倉

船倉の奥にもやはり多くの樽が置いてあった。ここにある樽はすべて空のようだ。J.B.はなにか凶器らしいものが見つからないかと樽を念入りに調べた。ひとつだけほかの樽よりも重い樽があったほかは，特になにもなかった。しかしそこでドアがあるのを見つけた。鍵が掛かっているようだった。先ほど手に入れた鍵を使うと，ドアは重い音をたてて開いた。

なかは真っ暗だった。しかし，J.B.はなぜかなかを非常によく見ることができた。

船倉を歩き進んで行くと，にわかに，太陽の光が見えてきた。J.B.は日の光がさしてくるほうに目をやった。そのとき，鳥に似たシルエットを見たような気がした。あれは!?

ACT4 疑惑

J.B.は，船長の息子を説得することになんとか成功し，彼を船から降ろした。

J.B.はもう一度アンドリューの部屋を訪れた。鍵が掛かっている。彼は自分の部屋にはいないようだ。今度はボブの部屋を訪れた。アンドリューはボブの部屋の前にいた。どうもボブの様子をうかがっていたらしい。J.B.に気がつくと，落ち着きのない様子を見せてこう言った。

「な，なんでもない……」

ボブの部屋を出て再びキリーの部屋に向かった。

「ハレルヤさん，私の愛用している斧が見つからないんです。ご存じありませんか」

もしかしたら，誰かがキリーの斧を持ち出してデビッドを殺し，その罪をキリーにかぶせようとしているのかもしれない，とJ.B.は思ったが，証拠はなにひとつない。

J.B.は船員の部屋を行き来し情報を集めた。キリーとアンドリューの姿が見えない。船内を探してみるとマストの上にキリーはいた。なにか情報はないかと聞くと，

「ええ，そういえば，今日はボブが見張り役なんですけど，金の壺がどうかしたとかいって，見張り役を今日だけ代わってくれというんです」

J.B.は，ボブがデビッドの部屋でなにか行動を起こすのではないかと思い，デビッドの部屋に向かった。部屋のドアを開けるとボブがいた。しかし，話しかけると，

「お，おまえには関係のないことだ。デビッドを殺した犯人でも探していろ」

と言うと，壺のようなものを小脇に抱え，ボブは部屋を出ていった。

ふさぎ込んでいる船長の息子を慰めるため，パーティはインコを求めて船内を東奔西走(写真左下)。そして結果はご覧のとおり(写真左)。心を開いた少年は，パーティの説得を受けて船を降りるのです。右下の写真は，これが噂のセントエルモなのです

ACT5 第二の殺人

J.B.は船長が用意してくれた部屋で，これまで集めた情報を整理していた。すると，けたたましい物音が聞こえた。船倉の奥で，なにかあったらしい。J.B.は船倉に急いだ。

船倉にはキリーがいた。

「ハレルヤさん，私は昨日ボブが金色の壺を海に捨てるのを見ました。なんでそんなことをしたんでしょうね」

J.B.はそんな話より，さっきの物音のほうが気になっていた。人の悲鳴のようにも聞こえたからだった。しかし，なにもない様子なので船倉から元の部屋に戻ることにした。ドアを開けて廊下に出ると，そこにはうつぶせになって倒れているボブの死体と彼の部屋の机の鍵があった。

犯人はいったい誰なのか。いちばん怪しかったボブは消えた。いまとなっては，一度捜査線上から消したキリーも怪しい。ボブが殺されたとき，キリーも船倉にいたからだ。また，アンドリューは姿をくらましたままだ。また，依頼主の船長だって捜査線上から外すことはできない。いったい犯人は誰なのか。この事件と「金色の壺」との関係は……。

最後にインフォメーション

いかがでしたか，このアドベンチャーミステリーの世界は。これを読んだだけでもゾクゾクしてくるでしょ。まだソーサリアンのこのシナリオに挑戦していない人は，ぜひこの真犯人を見つけ出してくださいね。ところで，もうすぐ（7月29日なのだ）このソーサリアンの新しいシナリオ5本が追加発売されるそうです。

どうやら今後，ソーサリアンにはこのような追加シナリオが引き続き発売される予定だそうです。さらに嬉しいことに，オリジナルシナリオを一般ユーザーから募集したりすることも予定しているそうなので，いまからネタを考えるのもいいかもしれませんよ。最後にもう一度ごめんなさい。先月載せてあった表1の見方に誤りがありました。表中の○印は単に順番を伏せてあるのではなくて，その魔法を使うのには○のところの要素が必要だという意味です。

では，また来月といいたいところだけど，ミュージック・モード頼むから誰か見つけてくれー。また，「シナリオ，ここがわからないコーナー」にもお便りください。

来月は，皆さんからのハガキを中心に，さらに傾向と対策を考えてみることにしましょう。

●第4のユニット2

サイコパワーで野望を砕け

Nagasawa Atsuhiro
長沢 淳博

WWWFが日本に乱入して来たって，そりゃ大変。なんとしてもこのチャンピオンベルトは渡すものか。と，ちょっと違うような気もするが，とにかくプロレス大ファンが作ったとしか考えられない楽しいアドベンチャーが再び登場したのです。

X1/X1 turbo用　5"2D版3枚組　7,600円
(2ドライブ専用)
データウエスト　☎06(968)1236

エッ？ なんですって。このゲームをこの私が解くの。どーれ，なになに，「第4のユニット2」。ツーってぇからには前作の続きものですな。アレッ，女の子が出てきたけど，まさかあの筋のアドベンチャーじゃないでしょうね。大丈夫だから1作目から試しに解いてみろって。ふーん，そいじゃ，チョチョイのチョイっと。

いやー，この1作目はなかなかアニメぽくって，プロレスっぽいアドベンチャーで楽しめますな。それじゃこの勢いで，「第4のユニット2」もいってみましょうか。

と，思ったらいきなりオープニングモードがありますね。ちょっと覗いてみましょうか（これはアドベンチャーの基本です）。

わたしゃ，本当いうと学園モノって好きなんです。前作は学校が舞台だったから今度もそうかと思っていたら，どうしてなかなか難しい話みたいじゃないですか。オマケニカタカナバカリノモジヲヨメッテイウノハトッテモツライ（そうでしょ）。要するに高級官僚が誘拐された事件を解決すりゃあいいんでがしょ。ようがす，この私が立派に解決してあげましょう。

待っていたよブロンウィン

この私は，ブロンウィンよ（いかん，ちょっと無理があるよな，これは。まっ，いいか）。こしなか博士，ストーリーの説明はいいから話を早くすませてちょーだい。えっ，エージェント？ なんだかよくわからないけど，さあ，行きましょ，セス。

——今回のアッシュの出番はここだけである。

ここで待ち合わせね。さっそく，あたりを調べてみましょ。アラ，いい車じゃない。ねぇ，セスもそう思うでしょ。えっ，車じゃなくてあの人を見ろって。まあ，男のクセにだらしないのね。バシバシ（兵士をたたいている音）。

——このブロンウィンは，“たたく”コマンドで平気で兵士でも壁でもバシバシたたく。なかには「ポムッ」っという音とともに，なにも起こらない場合もある。

あなたが，たきざわさん。サイボーグ計画って，なぁに。BSってブリヂストン，じゃなかったサイボーグのことじゃないの。

「ワッハッハッハッハ」

あ，その高らかな笑い声は黄金バット，もといダージリンにオレンジペコ。紅茶といえばアスリート。本当は全然関係ないダルジィちゃん。

「ふっふっ，そのとおり。あんたなんか目じゃないほど強いのよ」

アラ，いきなりドッカーン。「セスー，たきざわ中尉ぃー」

——それにしても，サイコブラストとはアニメアニメしている。

やってくれるじゃない，ダルジィ。こっちも必殺のサイコパワーッ！ 「スカッ」なにも起こらない。これじゃまるで紫醜罹じゃないの。ウッソー，なにその機械。そりゃ，掟破りの反則よ。レフリー呼んでやるからね。なんであなたのサイコブラストが通用して，この私のができないのよ。もう，プンプン。しょうがないわね，こうなりゃ肉弾戦しかないって。やーねー，私は嫁入り前の娘よ。

——というわけで，前作からお馴染みの戦闘モード。関節技やコブラツイストと，その技は決してほかのアクションゲームでは類を見ない。それにここでダルジィはブロンウィンのことをNo.4と呼んでいる。これで前作を知らない人でも，このゲームのタイトルの謎が少しは理解できる。

ハーハー，やっと勝ったわ，これで事件も万事解決っと，アレッ，なんで私が倒れなきゃならないの。なによ，勝っても負けても同じじゃない。結局はハンガーでUFOがどうしたっていうのよ。ちょっと，答えなさいよ。牢屋に入れたからって，この私がくじけるわけないでしょ。悲しいからって，マットレスなんて使わない。眠くたって，じっと耐えて考えるのよ。いつかは助けが来ることを祈って。

海のまん中無人島

この俺がたきざわ中尉だ。さっきはまったく，いきなりドカーンだもんな。なんてやつらだ。きっとこの手で必ず成敗してくれる。おっとそうだ，まずは地下2階の調査からだ。まずはカギを探して，うーんと次は……。ヤバッ，ダルジィだ。逃げろや逃げろ，スタコラさっさ。

ここはトイレか。そういえば最近便秘ぎみだし，ちょっとお借りしてと。ゴーッ，あー，スッキリした。そういえば，さっき

出たぁ，必殺の関接技攻撃

誰か来たような。まぁ，いいや。結局ここは2カ所だけか。じゃあ，地下1階へと行ってみるか。

とにかく調べまくろうぜ。ホラ，思ったとおりここにあるだろ。人の話はよく聞けよ。ついでにコンピュータも……。うーん，ほかに行けるところはないかいな。よしもう一度捜査続行だ。

なんと，ロスナンバーのレストア。女のことなら俺の出番じゃないか。中尉じゃ無理だって。てやんでぇ，こちとら江戸っ子でぇ，とにかくセーブでぇ。勝った勝った，おおっ，ここは収穫が多いなぁ。ここでこうなら次はあそこじゃないか，行くぞ。よしよし使えるぞ。ところで，これでなにを調べるんだ。ほうほう，ちゃんと照合装置まで付いてるじゃねぇか。よし，これで俺が……，なんでぇ，どうしてブロンウィンはいいんでぇ。

おおっとここで大尉の乱入か。俺は中尉だ文句あっか。

「カードのありかを教えろ」

セス，横から口をはさむんじゃねぇ。ほら，行っちまったじゃないか。今度はなにを調べるんだ？ サイボーグ計画と要人誘拐の関係か。うーん。

──実際，このゲームのストーリー展開はよく練られていて，重要なファクターである。

俺には関係ないね。それより，またあそこへ戻るんだ。それっ，あんな奴やっつけちまえ。ボカッ！ ほーら，これで新しい展開になっただろ。ハッハッ，俺の読みはズバズバ当たるなあ。

ここ，地下3階にはあれとこれがあるんだよなぁ。そしたらあのカギだってことはこの俺様にかかれば簡単。さて，戻ってみれば変な奴がいる。トヒニー・ユリト曹長か。なんだトイレ野郎だな。よろしい，ここは物々交換といくか。どうせまたここには戻って来るわけだからな。

オー，なんてこったい。ブロンウィン，君の秘密はこれだったのか。ロッカーのなかからとんでもないものを見つけてしまった。気にするな，ブロンウィン。人生，山あり谷ありだ。向こうの部屋ではサイボーグ計画の真相までわかってしまったぜい。

しかし，俺はいつからこんなにコンピュータに詳しくなったんだろ。ええい，ままよ。最後に残ったあの部屋に突進だぁ！

と，いう前に彼女に電話しとかなきゃ。これで準備OK。ユニットとコードを忘れないように。えっ，俺はムーバン・デルーゴ中尉なんかじゃないぜ。とにかく積年の怨み，一気にここで晴らすんでぇ。

あれがディフェンスシステム

どうも，おっちょこちょいのたきざわ中尉なんかには任せておけないわ。ここではこのカードを使うのよ。ほら，当たった。えっ，カウントダウン継続中？ いきなりなによ，これじゃあまるでストームじゃないの。

──どうやらこれにはX1のタイマを使っているらしい。

O・Cルームってなんか嫌な雰囲気ね。一応，これとあれともらっていきましょ。やっとこれですべてのアイテムが揃ったわ。ちょっと，たきざわ中尉，なに驚いてんのよ。エーッ，ウソー，この悪い連中ってこんなこと考えてたの。サイテー。なんてことでしょ。ポムッ。ああら，あんたなんか親でも兄弟でもないんだから，こっちを見ないで。私はこれから要人を助けに出かけるんだから。

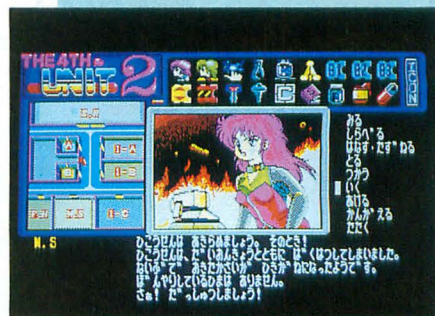
さあ，C＆Cって，どこかキャッチコピーみたいな部屋に行ってみましょ。ここは照合装置がクセものなのよ。セス，あとはたきざわ中尉が取り戻してくれるはずだけど，とにかく入りましょ。へぇ，ここでコントロールしてたわけ。あら，セス勝手にドタバタやってるわね。

──このあたりから，アニメが中心でストーリーが進む。そのうえ情報を全部集めるまでは「いく」コマンドを実行してくれない親切設計になっている。

これだけ聞き出せばもう十分。要するに，WWWFは世界を相手に自分だけチャンピオンベルトを手に入れようとしていたわけね。さあ，行きましょう。アレッ，なぜかこの私がもうひとりいる……。そう思った瞬間，ドッカーン，ドッカーンの大爆発。そしてゲームオーバー，となってたまるもんですか。もう一度たきざわ中尉に活躍してもらうわよ。そうすれば要人も救出できたし，いよいよ潜水艦で脱出よ。

そうは問屋が卸さない

さて，大混乱の様相を呈してきたドックからは実況中継でお届けします。なんとダルジィが突如リングに乱入してまいりました。おおっと，セスの声でブロンウィンが逃げようとしたー。でもダメです。どーやら敵に後ろは見せられないと判断したようです。ましてや「人間としてのプライドをかけても負けるわけにはいかない」とブロンウィンは叫んでいるー。か，感動です。過去にNo.4と呼ばれたこともあったブロンウィンが，いまは人間として敵に立ち向かおうとしています。

どうです，しっかりアニメしてるでしょ

果たして炎の海を無事脱出できるか

ついにダルジィと最後の決着を着けるため，両者が燃えるような闘志をみなぎらせ，リングで向かい合っています。どうやらブロンウィンのセコンドにはこしなか博士が付いている模様です。これは百人力の味方を得たようなものです。ブロンウィン，いけー，倒せー，やっつけろー。おおっ倒れたー，ついにダルジィが倒れました。息詰まるような熱戦にもようやくいま終止符が打たれようとしています。レフリーのカウントが入る。

「ワン，ツー，スリー」

やりました。さすがです，ブロンウィン。ついに勝利しましたー。

クーロンの謎は第3弾へと続く

いやー，それにしてもこのゲーム。なかなか面白いじゃないですか。アドベンチャーといっても操作はコマンド選択方式で，ストーリーを重視しているようだし。おまけにイヤミのないギャグは随所に見られるわで，ほんとに素直に遊べるアドベンチャーゲームです。

しかもこのあと，さらに続編も予定されているようなので，ブロンウィンの謎も徐徐に解明されていくことでしょう。ですから，ちょっと地味だけど，主流に流されないオリジナリティを持ったアドベンチャーゲームとして，この「第4のユニット」シリーズはこれからも成長していってほしいものです。

われら電脳遊戯民(1)

ゲームは僕らのキャンバスだ

Kuramochi Ryoichi
倉持 亮一

X68000の登場によりにわかに活気づいてきた最近のゲーム環境。これだけ我々を熱くするその要因はいったいどこにあるのか。機種の壁を越えその本質に深く潜入する「われら電脳遊戯民」が，いまここにスタートした。

ゲームは回る

去る昭和63年4月11日，ゲームメーカーであるセガ・エンタープライゼスが，体感ゲームシリーズ第8弾として「ギャラクシー・フォース」を発表した。

この体感ゲームシリーズには大きく分けて2種類あり，ハングオンやヘビーウェイト・チャンプなどのように操作を手元で行うのではなく体全体を使う“体が資本”タイプと，映画「トップ・ガン」の話題性に伴ってマスコミにも取り上げられたアフター・バーナーのように，実際にマシン（筐体）に乗り込み自機を操作すると，その方向にマシン自体が傾いたり回転したりして動く“三半規管を刺激する”タイプがあり，このとき発表されたギャラクシー・フォースは後者であった。

このギャラクシー・フォースを初めて目の当たりにしたゲーム雑誌の編集者たちの感想は，一様に「こりゃ，恥ずかしい」というものであった。なにしろこのマシンは，下部が電飾ギラギラ，ネオンサインしてるのに，上部はフレームとモニタ，操作系だけでプレイヤーの一挙一動が丸見え。マシンが前後に揺れるのはいいとしても，左右の移動操作に連動してそれぞれ最大335度まで回転するのだ。つまりゲームに熱中してマシンを左右に動かしていると，自分の体がマシンと一緒にグリングリン回転し，周囲に群がるギャラリーたちと思いっきり目が合ってしまったりするのだから，これは恥ずかしいに決まっている。

どのアーケードゲームにしても，初めてプレイしたときというものはアッという間に玉砕してしまって，残るものといえば自己嫌悪と羞恥心だけ。しかし，そのゲームの魅力がプレイヤーの血を沸かせたりするともうたいへん。「羞恥心と自己満足を計りにかけりゃ,金が出ていくゲームの世界」なのである(古いって，そりゃ失礼)。そしてそのゲームにある程度慣れてきて先に進めるようになってくると，“恥”が転じて“快感”となってくる。

結局，アーケードゲームに限らず自分の上手なプレイというのは人に見られたいものなのだ。しかしここで面白いのは，プレイヤーの意識として「見せたい」ではなく，「見られたい願望」がフツフツと湧き出てくることなのだ。ゲームセンターのテーブル型でプレイしていて，背後に人の視線を感じたりすると，次第に感情はエキサイトしてくる。しかし，顔面紅潮，指先ブルブルとなるかというと，さにあらず。外見はますます平静を装ってしまうのである。

さらに「平静を装う」意識はやがて「芸術的プレイ」願望へと成長し，「ここで敵がこう攻めて来るからかわしておいて，そこでこう撃つ」というように，ゲームの進行を把握して動きに一切無駄がなく，後ろで眺めている人間を「うっ……」と，うならせてしまう，そんなプレイを目指すようになるのだ。

実際，これまで流行ったゼビウス，グラディウス，ドラゴン・スピリットなどのように，難易度が死ぬほど高い（少なくとも私はそう思っている）シューティングにはこのような“芸術家”タイプのプレイヤーが存在するのだ。

しかしこうなってくると，もはやゲームセンターなるものは，個人がゲームを楽しむ場だけではない。ゲームの芸術家たちがその実力を発揮し披露する場なのである。

4畳半の東京ドーム

昭和61年12月にナムコが発売したファミコン用ゲームソフト，「プロ野球・ファミリースタジアム」はその後爆発的な売れ行きを見せた。しかもこれは一過性の流行ではなく，ドラクエシリーズと並んでファミコンソフト界のロングセラーとなったのだ。

しかし，なぜこれほどまでに野球ゲームが売れたのだろうか？　やはり野球に限らず，誰しも上手にプレイできればスポーツというのは楽しいものなのだ（これはゲームでも同じ)。2人プレイが可能なファミスタの場合だと，自分の行動とそれに対する相手のリアクションが加われば，勝負結果が状況によって毎回変化するので，ことさら面白くなる。

それに，実際に本格的な野球をしようとすると広い場所と18人のメンバー，そして2時間以上の時間と道具一式が必要となる。まさかタタミ6畳分のスペースに，18人も大の大人がひしめき合ってプレイするわけにはいかないし，9回裏までを30分で終わらせるなんてことは当然できない。そこで野球とゲームが好きな人間は，4畳半のスペースで18人分のプレイができ，おまけに30分もあればひと試合分くらいは消化できる，ファミコンゲームという“環境”を利用しようということになるのだ。

ファミスタ人気のあと，いくつか野球ゲームがファミコン用に発売された。いまもまだその傾向は続いているようだ。しかし，そのどれを取ってもまだファミスタを越えるものは出ていない。これはまたどうしたことなのだろうか。先にいいサンプルがあ

るのだから，迎撃するのは簡単だろうに，と思いつつ私はその理由を「ファミスタの適度なデフォルメ（déformer：つまり形を崩しちゃうこと）」と「適度な環境設定」にあると見た。

ファミスタ以降の野球ゲームは，ファミスタとの相違点を明確にするためにいろいろな“付加価値”を与えた。キャラクターをリアルにしたり，画面のアングルを変えたり，キャラクターの成長要素を加えたり，ゲーム中のイベントに凝ってみたりと，このままでは監督が審判を殴って退場したり，選手が起こした不祥事によってクビになったりするようになるかもしれない。まあ，これは実際の野球を観戦する側からすればよりエキサイティングでいいのだろう。やはり人に見せる試合というものはいくらかのショウアップが必要だ。

しかしこのような要素は，実際にゲームプレイする立場だとあまり有難くはない。キャラクターの設定がよりリアルだったり，イベントがしつこいぐらいに凝っていたりすると，かえって“自己投影”がしづらくなってしまうのではないか。やはり愛着がわくといった点では，リバイバーのような5等身キャラよりも，2等身や3等身でクリクリ目玉のキャラのほうが素直(シンプル)な分だけ，自己投影がしやすいということだろう。

結局，ゲームというのは“疑似環境”なのである。実際に東京ドームを使うわけにはいかないのだから，野球ゲームという疑似環境を利用するのだ。もちろんプレイヤーが東京ドームに入場するようにブラウン管に入ることなんてできないので（当たり前だよ，感電しちゃうよ)，自己投影という手段を使うのである。

しかし，この疑似環境にも自己投影しやすいものとそうでないものがある。たとえば遊園地というところはいろいろなアトラクションが用意されていて楽しいところだが，当然，入場料なるものを取られるので，遊園地が好きな人しか入らない。しかし，これが空き地であれば特になんの設備がない代わりに誰でも好きに遊べる。もしこの空き地と野球道具があれば，たちまち野球場となるのである。

つまり，自己投影しやすい疑似環境とはこの空き地のようなものである。そうして選手やダイヤモンド，スコアボードを用意してくれたのがファミスタというわけだ。やはり単純に野球を楽しみたいといった感覚でプレイするのは豪華なグランドよりも，質素な空き地が気軽で熱くプレイできる最適の場所なのだ。

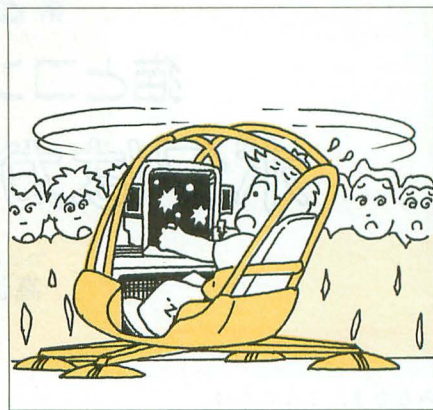
ゲームは小説より奇なり

ここでパソコンゲームの場合を考えてみよう。小説は文字情報しか与えてくれないから，読者はその文章からイマジネーションを働かせて自分だけの世界を想像する。たとえば「そこに美しい女性が歩いて来た」と書いてあれば，「ああ，きっと赤いドレス着て優雅に歩いているのかしらん」と，容姿はもちろん，着ている服や歩き方までを想像する。しかし，その想像した世界はその次に来る文字情報から必然的に修正を余儀なくされる。もしこのあとに「その女は腰ミノを付けて，踊りながら近づいてきた」と書かれていれば，ビックリして（そりゃ誰でもビックリするわな）自分のイマジネーションを根本から変えなければならない。しかし，この意外性が楽しさにつながってくるのだ。こうして小説は書き手と読み手のコミュニケーションのなかに成立している。

作家は文章を面白くしようと自分の知識や能力を最大限に発揮して文章を形成する。それに対して読み手は，より一層楽しく読もうと行間を読み想像力を働かせ，さらには先を予想し結局は裏切られる。この繰り返しはパソコンゲームでも同じだ。ゲームの作者は，常に楽しいものを作ろうと全身全霊を傾ける（そうじゃないのも一部いるようだが，そんな奴のことは知ったこっちゃない)。そしてプレイヤーはそれを受けて自分の思うところでリアクションを与え，時には作者の思うツボにはまって，あっさりと死んでしまう。しかし，このようなパソコンゲームの面白さは，こういった作者とプレイヤーとのぶつかり合いにあるのだ。

全二重キャンバス

ここまで疑似環境がどうちゃらとか，自己投影がうんちゃらなどという話をしてきたが，実際「ゲームってなんじゃろ」と考えるとキリがない。電動ドリルでも歯が立たないかもしれない。ゲームの世界はショウの舞台にもなるし，野球場にもなる。または戦場や雀荘，歌舞伎町のエッチな街角にだってなるわけだから，さまざまな可能性を秘めている。しかし，アーケードからパソコン，ファミコンを問わず存在するゲームにも大きな共通点がある。それはどれをとっても“自己表現の場”であるということだ。

ゲームセンターで芸術的プレイを披露しているときも，そこにはアピールしている

自分がいる。ファミスタで遊んでいるときは選手の1人ひとりに乗り移って熱中している自分が，そしてパソコンゲームでは送り手が創造した世界の中で自分自身を演出してくれる空間をさまよっている。こうして送り手も受け手もゲームという環境のなかで自分というものを表現しているのである。

ゲームが自己表現の場であるとするならば，ゲームは文学界やスポーツ界，果てはアイドルやバラエティ番組をも含めた放送の世界などと共通する部分も多いのかもしれない。えっ，話が飛躍してきたって？アイドルの場合を考えてごらんなさいな。自己の存在をアピールするためには，ルックスや歌のひどさ(これは逆か)，スットンキョーな性格などを前に出して自己表現をしないと，有象無象のひとりよがりで終わってしまうのだよ。

こうして自己表現を主体にこれまで蓄積された遺産，そしてこれから構築されるだろう新しい世界。これはもうもはや我々だけがかいま見ることのできる“文化”である。

ゲームというのは言い換えれば誰でも自由に描くことのできるキャンバス（当然，画布のことである。断じて大学構内だとかコクヨのノートのことではない）みたいなものなのだ。真っ白なキャンバスの上に送り手が面白いと思う下絵を描き，プレイヤーは自分なりのカラーに描き変えていくのだ。このとき文化を創造することをどこかに考えて描くことをすれば，キャンバスの上にはゴッホやピカソも真っ青の芸術作品に仕上げることだってできるはずだ。

その逆にゲームを単なる幼稚な遊びだとしか考えていなければ，保母さんから「もっとがんばりましょう」のハンコしかもらえない子供のお絵描きにしかならない。そしてその絵の作品がどちらに転ぶかは，当然，送り手と受け手の両方の感性にかかってくるのである。

第**26**回

猫とコンピュータ

ボクはかぐや姫？

Takazawa Kyoko
高沢 恭子

物語はいつも突然始まります。高沢家の白猫ホンニャアも，とうとつに気を変えるのが得意です。久し振りに古巣へ戻ってきた恭子さん，今回は彼の気まぐれにかなり忙しい思いをしたようす。とはいえ，フルメンバーが揃うとやっぱり無敵の「猫とコンピュータ」です。

みなさま，こんにちは。

そして，もしかして，白猫のホンニャアを覚えていてくださったかた，ごぶさたいたしました。

ホンニャアはとうとう転勤で東京に引っ越してきました。今年のはじめのことです。トオルは6年生の3学期でした。もちろん転勤したのはホンニャアではなくて，彼の一番信頼しているトオルのパパです。

丸の内の本社はこれで3度目だし，転勤も8回目くらいですが，今度の引っ越しだけは少しようすが違いました。ホンニャアがいるからです。

ホンニャアはふしぎなことに，転勤が内定した日の朝早く，夫の電話がくるよりも前に行方をくらましました。

私たちはそれまでにも，転勤になった場合のホンニャアのことを，何度も真剣に考えました。彼は，この家を含めたまわりの自然のなかで生き生きとすごしているのだから，ほかの環境に持っていくのはかわいそうだと，3人ともいつも思っていました。

申し遅れましたが，わが家は私と夫と息子のトオルの，3人家族です。

あのワガママ猫を，誰かに代わって飼ってもらうなんて，とてもあり得ません。でも一緒に連れていくことは，ホンニャアがほんとうに喜ぶことなのでしょうか。

考えはいつもここで行き詰まってしまい，結論はそのたびに見送られてきました。

ホンニャア，月に帰る？

ところが，いざ「ホンニャアのいる引っ越し」が現実のものになったとき，肝心の議題の主が見あたらなくなってしまったわけです。おかげで心配なのもさることながら，引っ越しの準備はいままでになく混沌としてきました。

引っ越しをするときは，一番はじめに住むところを決めます。通勤のこと，通学のこと，毎日の生活のことを考えて，できるだけ望ましいところを選びます。

住むところが決まると，通う学校も決まるし，家具や身のまわりのものの要不要も予測できるのです。

ただし，そこに動物が同居するとなればやや話が違ってきます。以前の東京暮らしのように，マンションというわけにはいきません。一戸建ての家を見つけるのなら，2階もあることだろうし，いまの家具をそのまま運んでも納められそうです。

というぐあいに，面倒ながらも浮き浮きした環境設定が楽しめるはずなのですが，8日たっても，9日たっても，ホンニャアは戻りません。もしこのまま彼が現れなければ住居はマンションでもかまわないことになり，計画は大幅に変わってくるのです。

こうして，私たち一家は，猫1匹にだいじなカギを握られてしまいました。

ホンニャアが消えてしまった話を聞いた近所の人たちは，「そりゃあ，ほんとにリコウな猫なんですよ。家族に迷惑がかからないように，身を引いたんですね」と，ほめちぎりました。

そのときはもう10日以上たっていましたし，同じようなことを考えていた私たちにとって，こんなつらい「ホンニャアへのほめ言葉」はありませんでした。

あのキャットフードしか食べられないホンニャアが，自分から家を出ていくなんて考えられないのですが，近所の人の言葉でなんだかホンニャアとの別れが決まってしまったように感じ，急に悲しさが増してきました。

そうはいっても，引っ越しをいつまでもためらってはいられません。夫は決められた日から新しい任務につくのです。

「住まい」は都内のマンションということで地下鉄T線沿線に探してほしいと，会社に手配を依頼しました。

そうと決まったら，ホンニャアはもう「帰らぬ猫」です。私たちもあきらめの気持ちをはっきりさせなくてはなりません。

3人であれこれ話をしました。

「ホンニャアが自分で選んだ道なら，それが彼の幸せなんだ。それに，動物は野生のカン，特に猫は人間にはもうない神秘的でふしぎな能力に恵まれているから，きっと飼い猫でなくなっても元気に生きていくだろう。楽しい日々だったけれど，やはりホンニャアと私たちは別の世界に住む者同士で，ひとときの出会いはこれで終わりを告げたのだ……と。

こうなってくると，あの，おなかを天井に向けて眠り呆けていた姿[1]はすっかり忘れられ，ホンニャアのイメージは月の世界に帰っていく「かぐや姫」みたいに，神々しく輝いてくるのでした。

さて，おおかたの日用品は，ほとんど段ボール箱に姿を変えて部屋の中に積み上げられ，残っているものは引っ越し当日に梱包してもらうカラのタンスや食器棚，食卓やイス，あとはほんの数日ぶんの衣類と食器類になりました。汚れた毛皮なんかが決して残っているはずはないのです。

夫も，勤務のために，ひと足さきに仮の宿舎である都内のホテルに移っていきました。………………

さぁ，たいへん!!

汚れた毛皮は，その日食卓の下で発見された。

毛皮は自分で動いて，手や体をなめまわしていた。それは汚れた「猫(！)」だった。

ホンニャアは一気に月から墜落して，私の目の前にいた。故障した宇宙船の噴煙でも浴びたのか，みすぼらしく黒ずみ，かぐや姫の名残りはどこにもなかった。

「もしもし，パパ，たいへん！　帰ってきたのよ，ホンニャアが」

言っている自分の声が信じられないようなこの現実。でも，やっぱり彼が私たちをもとめて帰ってきたことが満足だった。

「そうかぁ……」

夫の声も嬉しいような，とまどったような響きだ。ただ喜んでいるわけにはいかないから，ほんとは2人ともちょっと動揺している。

まず，いそいで住まいをマンションから

一戸建てに変更するよう頼み直さなければならない。無理な頼みとは承知しているけれど，マンションで猫を飼うことはできないし，ホンニャアだってかわいそうだ。

当然，引っ越しも少し延期しなければならないことになる。トオルの転校が遅れるのが飼い猫のためだなんて，誰に言えるだろう。たくさんの恥を我慢して，この際いろんな決断を一度にやらなければならない。

そうだ，ホンニャアはどうやって運ぶのだろう。

会社の引っ越しはいつも「日通」こと日本通運さんがやってくれることになっている。

「あの，猫…なんかいたらどうなるんでしょうか」

と打ち合わせにみえた日通の担当の方に聞いてみたら，

「どこの日通でもOKというわけじゃないんです。営業所の判断にまかされてるんですよ」

「やっぱり，あんまり歓迎じゃないんですね？」

「でも，1匹だけでしょ。なんとかオリに入れて，運転台のそばにでも乗せていくように言ってみましょう。たいした距離じゃありませんしね」

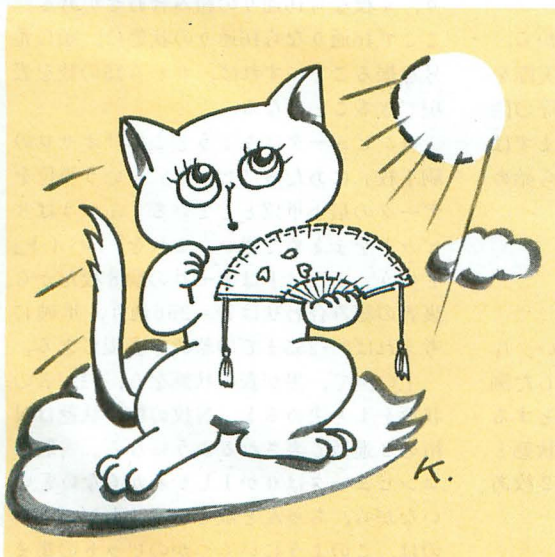
引っ越しの日取りも決まらないうちに，猫のほうの相談が先に進んだ。

後学のためと考えて，いまを盛りの宅配便や引っ越し産業のいくつかに，猫やペットの運搬について聞いてみた。

まずは，黒い親猫が子猫をくわえて運んでいるマークで大繁盛の「ヤマト運輸」。

「モシモシ，あの，引っ越しも引き受けてくださるんでしょ？」

「はい，お受けしています」

と若い女性の声。

「あの，そういうときに動物も運んでいただけるんでしょうか？」

「動物っていいますと…なんですか？」

「あの，猫なんですけど」

「あ…ちょっとお待ちくだサイ……」

と語尾を上げながら，その人は電話のそばを離れて30秒ほどすると，

「あのぉ，申しわけないんですけど，動物は，その，ちょっと……扱わないことになっています……すみません」

「あ，そうですか。やっぱり自分で運ぶんですね。わかりました」

続いて8ばかり並んだ電話番号の，「ダック引越センター」，

「申しわけありません……わたくしどもの会社では，生きものはお受けしないことになっております。はい，万一のときに責任が負えないことになりますので……」

ついでにもうひとつ，胴のながぁーいワン君，ダックスフンドがシンボルの「フットワーク」。

「動物は何が起きるかわかりかねますので，お断りしているんですけど……ほんとにすみません」

もっともな理由だし，商売だからなんでも運んでくれるというのは，ちょっと甘い考えのようだ。きっとほとんどの人は自家用車で運ぶのだろうけれど，大きな動物やたくさんいるときはどうするのかな。

でも，「猫」も「アヒル」も「犬」も，みぃんな動物を運んでくれないのが面白かった。

白猫便

それからあまり時間をかけずに，なんとか一戸建ての家をみつけてもらえた。小さいけれど，ホンニャアがホッとできるくらいの庭もある。大手町から16分，駅から4分のところで，大きな団地群と向き合っている。

ホンニャアは唯一の頼りの「日通」さんを予定していたが，直前になって，研修旅行帰りだという私の兄（狛江のアニキです）の車で運んでもらえることになった。

「あのイナイイナイバーをする猫[2]か？ オレ，責任は持たないヨ」

「私たちも一緒に乗せてもらうから，なんとかなると思うの」

兄は引っ越しの当日，荷を積んだトラックが出ていくのと入れ替わりに，約束どおり来てくれた。

この日は朝から波乱に満ちていた。

ほかのことはすべて順調でなんの心配もなかったのだけれど，ホンニャアを「車に乗せてからの問題」の前に，「乗せる問題」がなかなか深刻だったのだ。

出発の予定時刻に，ホンニャアが遊びや散歩に出かけないとも限らない。いや，出かけることのほうが多いくらいだ。なんとか，ちょうどそのころ昼寝の態勢にはいってくれないものか。

トオルの転校よりも，パソコンよりも（このごろでは夫ばかりでなく，私にもだいじなパソコンです），何よりもホンニャアに重点の置かれた引っ越しになってしまった。

ところが，兄の車が来てからのホンニャアはなんともふしぎだった。

それまでは外にいたホンニャアが，部屋の中で正座して待つようなかっこうになり，あの，お医者さんに連れていかれるときに入れられる大キライな金具のオリにも素直に入ったのだ。

車が走り出してからも，いつものように狂ったように暴れることもなく，2時間半を静かに耐えてすごした。平日で道路の流れが順調だったことも幸いして，私たちは思いがけなく，穏やかにホンニャアの移動を済ませることができた。

一度は私たちと別れるつもりだったホンニャアが，考え直して帰ってきたように見えるのも，ほんとはただの気まぐれなのか，誰にもわからない。

この，ホンニャアの天性のパントマイムに，これからも魅了され，振りまわされながら，私たちは暮らしていくことだろう。

トオルは，3つ目の小学校も楽しく悠々とすごし，友だちにも恵まれて，この4月からは無事，中学生になった。徒歩50秒，わが家の目の前にある公立中学校で，ますます充実の毎日のようだ。

住まいが都内になった一番の利益は，夫の通勤よりパソコン通信かもしれない。都内のネットなら1時間アクセスしても200円くらいだ。これは最高にありがたい。

でも，ついでに新宿の実家も近くなってしまったので，おばあちゃんのお説教もひんぱんに聞かされなければならない。

これだけが，ちょっとつらいかな？

さて，ホンニャアの新しいナワバリはどんなふうに展開するだろうか。

注 1） これがホンニャア得意の寝相です。
2） わが家では，かつて彼に「イナイイナイバー」を仕込みました。ホントです。

※この連載は，昨年7月号まで25回にわたり本誌に掲載されました。ホンニャアと高沢家とパソコンには，ほかにもいろいろなエピソードがあります。詳しくは7月下旬発行予定の『猫とコンピュータ』でどうぞ。（編集子）

特集1

真夏の夜の数値演算

コンピュータにおける数値表現……28
連立方程式は行列でいこう……34
iがあるからむずかしい……42
とんでもなくデタラメな話……47
そこにπがあるから……51
超応用グラフィック──歪められた光……61
超応用AD PCM──音の数学……64
数値演算プロセッサの活用──FLOAT3+.X……72

数値演算と聞いて，なにやらドキドキしてくるという人はすでに演算の魔力にとりつかれた人かもしれません。そうでなくとも，数値演算という言葉がことのほか刺激的なのは，多くのユーザーがその重要性を知りながら，同時にうんざりするような数式をも思い浮かべるからでしょう。しかし，数値を扱うことは重要です。たとえば，私たちがBASICでなにげなく使っている実数もコンピュータの内部では「浮動小数点」と呼ばれる概念のもとに扱われています。マシン語の世界に入っていこうとするならコンピュータが扱うことのできる数値の意味を理解しなければならないでしょう。さらに，グラフィックやサウンドなどの用途でも，その華やかな舞台の裏側に一歩足を踏み入れると必ず待ち受けているのが難しい数学の知識なのです。
さて，今回は数値を扱うさまざまなテーマを集めてみました。コンピュータの計算能力には人知を超えるものがあり，それゆえに解決可能となった問題もあります。またパソコン程度では夜を徹して計算しなければならないものもありますが，それも時間さえかければ必ず解けるということです。夏の夜こそパソコンをフルに稼働させてみてはいかがでしょう。

コンピュータにおける数値表現

Murata Toshiyuki
村田 敏幸

パソコンで数値演算をやる場合には数学の常識では起こりえないことも往々にして発生します。その多くは数値表現に関するものですが，なぜそのようなことが起こるのか，どうすればそれを避けられるのか，といった数値演算の基礎の基礎から始めてみましょう。

計算を考える

電子計算機というほどだから，コンピュータは計算が得意だと思われがちだ。が，もともとコンピュータは特定範囲での整数の加減算，せいぜい乗除算くらいまでの演算機能しか持っていない。そのコンピュータがどうやって大きな整数や複雑な実数の演算をしているかという疑問は，我々人間が計算するときのことを考えればすぐに氷解する。

たとえば，人は数十桁もの数の和を即座に求めることこそできないが，下の桁から1桁ずつ足して，繰り上がって……という手順で計算することはできる。また，掛け算についても，九九（つまり1桁の整数同士の掛け算）程度を覚えておけば，どんなに大きな数だろうと掛けあわせることができる。さらに小数の演算をするときは，小数点の位置を調整する以外は，整数とまったく同じように計算することができる点を指摘しておく。結局，「限られた範囲の整数の四則演算」を知っていれば，大きな整数の演算や実数演算を行うことは可能なわけだ。

以下，かなりシステムよりの視点から，パソコンで行われている数値演算の実際を紹介し，高級言語で数値演算を行う際の問題点を明らかにしていこうと思う。まずは，数値の内部表現形式についての話から始めよう。

2進数の基礎

オセロゲームの駒を想像してほしい。片面が黒で，もう片面が白い円盤状をした例のやつだ。この駒が1枚だけあったとすると，駒の状態は白が上になっている状態と，黒が上の状態の2通りがある。駒が2枚あるとすると，考えられる状態は，

黒　黒
黒　白
白　黒
白　白

の4通りだ。同様に3枚の駒があれば8通り，4枚なら16通りの組み合わせがある。ここで16通りなら16通りの状態に，順に番号を振ることにすれば，0から15の数を表現できることになる。

コンピュータはちょうどこの「オセロの駒1枚」にあたる「ビット」という単位をデータの最小単位としている。ふつうは8ビットをまとめて扱い，これを「1バイト」という。1バイトはオセロの駒8枚だから，裏表の組み合わせは$2^8=256$通り，単純に考えれば0～255までの整数を表現できる。

上の例で，黒が表の状態を0，白が表の状態を1と決めると，N枚の駒の状態はN桁の2進数で表されるようになる。一般にコンピュータは0か1しかわからないといいながら，ちゃんと数を扱うことができるのは，このようにいくつかのビットの集ま

り（ビット列）を2進数とみなして扱っているからにほかならない。

負を表す4つの方法

続いて負の数の扱いを考える。もっとも理解しやすいのは「絶対値表現」と呼ばれる方法だ。これは数値ビット列のほかに，もう1ビット，符号ビットという目印を用意し，これが0なら正の数，1なら負の数と決めることで負の数を表現する。この方法なら8ビット（7ビットの絶対値＋符号ビット）用意すれば－127～＋127の数を表現できるようになる。ふつうは最上位ビット（2進数で考えると最上位桁）を符号ビットとし，

00000001

なら＋1，

10000001

なら－1というように考える。

絶対値表現はわかりやすいのだが，いくつかの欠点があるので，あまり使われることはない。まず，下に示すように0の表現が＋0と－0の2つできてしまう。

00000000

10000000

また，演算をするときに常に符号を調べて処理を割り振らなければならないので，手間がかかるという問題もある。

負の数を表現する2番目の方法は「ゲタばき表現」といわれるものだ。この方法は任意の数を0とみなして，それより1大きい数を1，1小さい数を－1というように決めることで負の数を表現する。別のいい方をすると，ある定数（この定数をバイアス値という）を足すことで負の数を正にして扱う方法ともいえる。たとえば，

10000000

は正直に読めば128だが，これを0だと決めれば，それに1を足した，

10000001（ほんとは129）

を＋1，1を引いた，

01111111（ほんとは127）

を－1と考えることもできるわけだ。

ゲタばき表現は特殊な用途には使われるが，やはり一般的ではない。だいたい，

00000000

が0にならないのは不自然に見えるし，正の数だけを扱っていた場合と統一がとれていないのも問題だ。

第3の方法は「1の補数表現」で，ビットをすべて反転したものを負の数として扱うやり方だ。一例をあげると，＋1は，

00000001

だから，－1は全ビットを反転して，

11111110

というように表される。この方法はハードで行うのが簡単なので，一部の大型コンピュータでは採用されているようだが，やはり0が，

00000000

11111111

の2種類できてしまうなどの欠点がある。

4番目は「2の補数表現」で，これがもっとも一般的な方法だといえる。まず，

00000000

が0，それに1を足した，

00000001

が1となり，ここまでは正の数だけを扱っていたときと同じだ。で，－1は0から1を引くことで求める。実際には0から1を引くことはできないわけだが，最上位桁のもうひとつ上に1があるものとみなして無理やり引いてしまう。－1が求めたければ，

```
    (1)00000000
 -)      00000001
 ---------------
          11111111
```

となる。

この方法では期せずして，最上位ビットが符号ビットの役割を果たすようになっている。したがって，8ビットの場合のもっとも大きな数は，

01111111

で127，もっとも小さな数は，

10000000

で－128となる。

いよいよ実数表現

今度は実数の表現方法を探ってみる。まず2進数で小数を表す方法だが，これは次のように考える。

10進数で表された数，仮に123は，

$1\times100+2\times10+3\times1$

の意味だった。この式は，

$1\times10^2+2\times10^1+3\times10^0$

と書くことができる。これは各桁がそれぞれ10の「桁数－1」乗の重みを持っていることを意味している。これを拡張して考えると0.123は，

$1\times10^{-1}+2\times10^{-2}+3\times10^{-3}$

というように表され，小数点以下の各桁は10のマイナス「小数点以下の桁数」乗の重みを持っていることがわかる。

上の考え方をそのまま2進数にあてはめてみると，

10.01B（～Bは2進数を表す）

は，

$1\times2^1+0\times2^0+0\times2^{-1}+1\times2^{-2}$

と考えることができるだろう。このように2進数では小数点以下の各桁は2のマイナス「小数点以下の桁数」乗の重みを持っている。具体的には小数点以下1桁目は0.5の重み，2桁目は0.25の重み，以下桁が増えるに従って重みは半分になっていく。結局例に挙げた10.01B を10進数に直すと，

$1\times2+1\times0.25=2.25$

となる。

2進数での小数の表し方がわかれば，簡単ながら実数の内部表現を決めることができる。たとえば，整数部，小数部をそれぞれ8ビットに固定して，

1101110.11101111B

を，

01101110B　11101111B

の2バイトに分けて格納するようにすればよい。このような方法を「固定小数点表現」という。小数点以下が長くて入りきらない場合は切り捨ててしまうか，それがいやなら格納できる範囲のすぐ下の桁を「0捨1入」することになる。

いま例に挙げた固定小数点表現は，見方を変えると実数を256倍して16ビット整数に格納したような形式をしている。そのため，加減算は整数の演算とまったく同じ処理が使え，手軽に実数を扱うことができるという利点がある。

しかし，実際には固定小数点表現はあまり用いられることがない。それというのも，場合によっては有効桁数が非常に小さくなってしまうためだ。一例を挙げる。

0.0000110011101B

という有効数字2進9桁の数を先の固定小数点表現に変換することを考える。小数点以下が8桁を超えるので9桁目で丸めて（0捨1入して），

表1　整数の表現

10進数	2の補数	1の補数	ゲタばき	絶対値
127	01111111	01111111	11111111	01111111
126	01111110	01111110	11111110	01111110
:				
2	00000010	00000010	10000010	00000010
1	00000001	00000001	10000001	00000001
0	00000000	00000000	10000000	00000000
-1	11111111	11111110	01111111	10000001
-2	11111110	11111101	01111110	10000010
:				
-126	10000010	10000001	00000010	111111110
-127	10000001	10000000	00000001	111111111
-128	10000000	--------	00000000	---------

注）ゲタばき表現のバイアス値は80H（10進の128）

00000000.00001101B
となる。見てのとおり，小数点以下が入り切らなかったために非常に多くの情報が失われてしまっている。

ここで注目してほしいのは，上のほうの桁が余っている点だ。この余っているビットをも有効的に活用するためには，小数点の位置を固定ではなく左右に動かせるようにすればよい。すると，同じ16ビットの領域でも，

0000.000110011101B

というようにゆとりを持って格納できることになる。これが「浮動小数点表現」の考え方だ。

浮動小数点と正規化

浮動小数点表現は数を指数部と仮数部に分けるところに特徴がある。10進数でいうと，ある数を，

$X \times 10^{Y}$

の形式で表したときのXを仮数部,Yを指数部という。また，実際にコンピュータで使われる2進数での浮動小数点表現は，

$X \times 2^{Y}$

の形式で表されることになる。たとえば，仮数部16ビットの浮動小数点表現で，10進の0.5を表してみると，

$.1000000000000000_{B} \times 2^{0}$

となる。ここで指数部を増減することが，小数点を左右に動かすことにあたる。

ところで，0.5は上に示したように，

$.1000000000000000_{B} \times 2^{0}$　(1)

と表す以外にも，

$.0100000000000000_{B} \times 2^{1}$　(2)

$.0010000000000000_{B} \times 2^{2}$　(3)

$.0001000000000000_{B} \times 2^{3}$　(4)

などなどの形で表すこともできる。が，有効桁数が最大になるのは(1)のとき，すなわち最上位ビットが1のときだ。そこで最上位ビットが1になるように指数部を調整する操作が行われ，これを「正規化」という。(2)～(4)は正規化されて(1)の形となる。

正規化は有効桁数を確保する以外にも面白い効果をもたらす。

1) 正規化された数は最上位ビットが1であることがわかっている
2) 常に正規化すると決めておけば，最上位ビットは常に1になる
3) 常に1なら省略しても情報は失われない（意味は通じる）
4) というわけで，1ビット分ケチることができるようになる

このように最上位ビットを省略して，記憶領域を節約することはよく行われ，これを「ケチ表現」という。浮いた1ビットは別の用途に使うなり，指数部に回すなりしてもよいわけだ。

ケチ表現を使うときに注意しなければならないのは，0の扱いだ。0は浮動小数点表現で表しても全ビットが0になる。どうやっても最上位ビットが1にならない唯一の例外だ。ある数を正規化した結果，最上位ビットだけが1となるような場合に，ケチ表現を使って最上位ビットを省略すると0との区別がつかなくなってしまう。

これを避けるために，0にだけ特別な目印をつけることを考える。具体的には，指数部がある特定の値ならば0というように決めればよい。これにより，指数部の範囲が少し狭くなるわけだが，0を区別するためにはしかたのないことだろう。

なお，ここでは小数点以下第1位が1になるように正規化する例を示したが，1の位が1になるように正規化する形式もある。その場合0.5は，

$1.000000000000000_{B} \times 2^{-1}$

のように正規化される。

パソコンの数値表現

そろそろ理屈っぽい話を聞いているのにも飽きてきただろうから，実際に身近なパソコンで使われている数値の内部表現がどうなっているかに話題を移そう。図1にX1，MZシリーズのBASICで使われている整数，単精度実数，倍精度実数，ならびにX68000で使われる整数と実数の内部表現フォーマットをまとめておいたので，この図を見ながら以下を読んでほしい。

最初に整数だが，X1，MZは16ビット(2バイト)，X68000は32ビット（4バイト）という違いこそあるものの，どちらも2の補数表現による符号付き整数であることに変わりはない。ここまで読んでくれた読者にはもう説明はいらないだろう。

ビット数の違いは扱える値の範囲に表れ，16ビット整数は−32768～＋32767，32ビット整数では−2147483648～＋2147483647までの数を表現できる。また，X-BASICにあるchar型変数は無符号8ビット整数で，0～255の範囲の値を格納することができる。

実数はどのマシンでも浮動小数点表現で表されている。X1，MZでは5バイトの単精度実数と8バイトの倍精度実数があり，X68000のX-BASICでは8バイトの倍精度実数だけがある。先にX1，MZのほうを解説してしまおう。

まず，仮数部は単精度31ビット，倍精度55ビットの絶対値表現であり，別に用意された1ビットの符号ビットにより正負を決めるようになっている。また，仮数部はケチ表現で表されているから実質的な仮数部の桁数は上に示したものより1桁ずつ多くなっている。さらに，正規化は小数点以下第1位に1がくるようにする方式が採用されている。

もうおわかりのように，倍精度実数のほうが単精度実数よりも演算精度が高いのは仮数部により多くのビット数を割いているので多くの有効桁を得ることができるからだ。

指数部は単/倍精度ともに8ビットゲタばき表現となっており，X1，MZでは80H(10進でいうと128）のバイアスがかかっている。ただし，前で話した「0を表す特別な指数の値」として0が使われているので，有効な値は1～255，バイアス値を引いて−127～＋127となる。

仮数部は絶対値表現なのに指数部ではゲタばき表現が用いられているのは「ひとつの数の表現中に2つの符号ビットがあるのは好ましくないからだ」というもっともらしい説明を聞いたことがあるが，0の扱いとの兼ね合いでこうなっているのではないかという見方もできる。

「0を表す特別な指数部の値」は0にするのが直感的でわかりやすいのは確かだろう。しかし，たとえば0.5は浮動小数点形式では，

$.100\cdots\cdots \times 2^{0}$

となるように，指数部の0を勝手に使うわ

表2　数値の表現可能範囲

◎X1，MZ	
整数	−32768～＋32767
単精度	$\pm 2.9387359 \times 10^{-39} \pm 1.7014118 \times 10^{38}$，および0
倍精度	$\pm 2.938735877055719 \times 10^{-39} \pm 1.701411834604692 \times 10^{38}$，および0
◎X68000	
INT	−2147483648～＋2147483647
CHAR	0～255
FLOAT	$\pm 1.1125369292536 \times 10^{-308}$～$\pm 3.5953862697246 \times 10^{308}$，および0

けにもいかない。その点ゲタばき表現にすれば，0乗と「0を表す指数部0」とが区別できるようになり，めでたしめでたしというわけだ。

X68000の2つの数値表現

続いてX68000で使われている倍精度実数のフォーマットを見てみよう。X68000には実数の表現方法が2系統ある。ひとつは俗にシャープフォーマットと呼ばれる形式であり，もうひとつはIEEE(アメリカ電気電子学会。アイ・トリプルイーと読む)の規格に沿った形式だ。X-BASICの最初のバージョンでは実数の表現はシャープ形式だけだったが，バージョン2.00になった時点で数値演算はドライバで行われるようになり，ドライバを差し替えることでシャープ形式とIEEE形式を自由に選べるようになった。

2つの形式を見比べてみると，フィールドの順序が異なるだけで仮数部や指数部のビット数に違いはない。だから，どちらの形式を使っても表現できる数値の範囲や演算精度は変わらない。仮数部は52ビットで1の位に1がくるように正規化される。指数部は11ビットであり，3FF$_H$のゲタばき表現となっている。また，やはり0は指数部を0にすることで表す。

ここで，さきほど話したX1やMZで使われている倍精度実数のフォーマットと比較してみると，指数部に3ビット多く使っている分，表現できる数値の範囲は広くなっていることがわかる。逆に仮数部は3ビット少ないので精度は劣ることになる。X1やMZの倍精度実数のほうがX68000より精度が高いというのはちょっと意外な結果だろう。

ところで，まったく精度の同じ2つの形式が用意されているのは不自然に見えるかもしれないが，これには初期バージョンとの互換性を維持し，同時に標準的なIEEE規格も使えるようにすることで他機種との互換性をも得，さらに浮動小数点演算プロセッサを使えるようにする狙いがあってのことと思われる。

IEEEと浮動小数点演算コプロセッサ

浮動小数点表現のフォーマットには「絶対にこうしなければならない」という決まりはない。そのため，これまでは機種やBASICなどのソフトによってバラバラの形式が使われてきた。異機種間でのフォーマットの互換性は「あったほうがいい」程度のものかもしれない。が，少なくとも，同じ機種上ではどのソフトも同じ形式に統一されていたほうが好ましいのは確かだ。これにより種々のソフト間でデータを共有できるようになる。もちろん，異機種間でも互換性があればなおよいわけだ。

若干あてずっぽうだが，だいたいこんなわけで数値表現の統一化が望まれるようになり，数年の歳月をかけて1985年，この世界での立法府ともいえるIEEEによって，ようやく浮動小数点表現形式の規格が確立された。規格ができても，それに従わなければ意味がないわけだが，IEEE規格に沿った数値演算コプロセッサの普及に伴いIEEE規格は広く受け入れられるようになっ

図1　浮動小数点表現形式

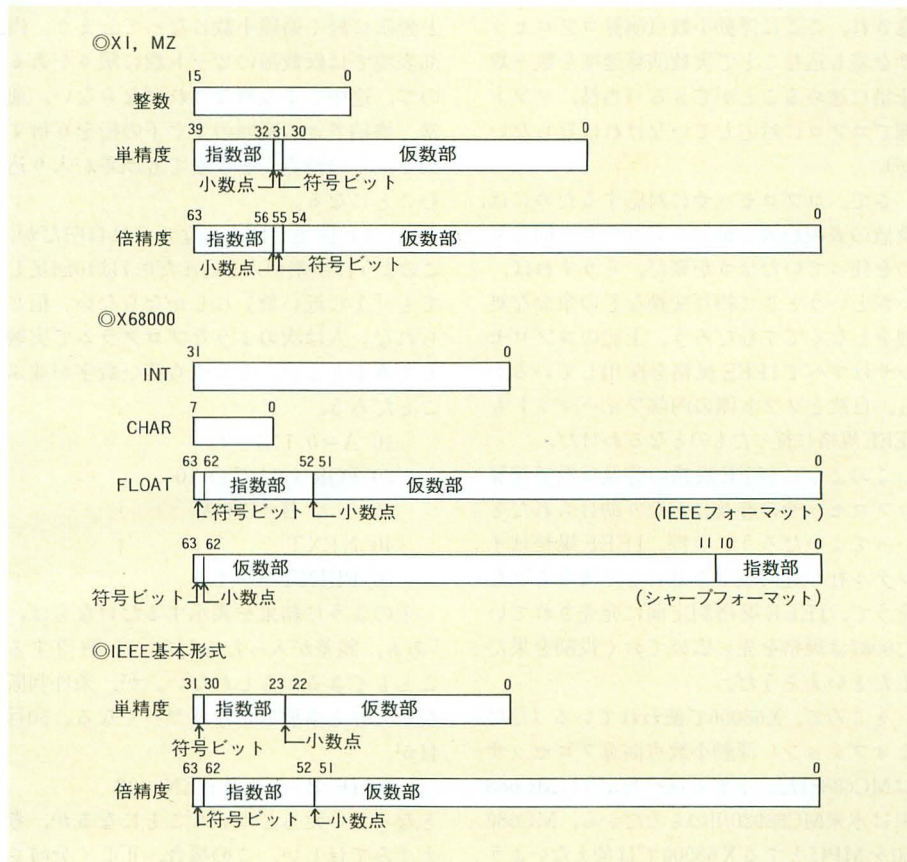



図2　浮動小数点表現の例

```
◎X1，MZ　単精度
+1.0  10000001 00000000 00000000 00000000 00000000
+0.5  10000000 00000000 00000000 00000000 00000000
-0.5  10000000 10000000 00000000 00000000 00000000
+0.1  01111101 01001100 11001100 11001100 11001101
 π    10000010 01001001 00001111 11011010 10100010

◎X1，MZ　倍精度
+1.0  10000001 00000000 00000000 00000000 00000000 00000000 00000000 00000000
+0.5  10000000 00000000 00000000 00000000 00000000 00000000 00000000 00000000
-0.5  10000000 10000000 00000000 00000000 00000000 00000000 00000000 00000000
+0.1  01111101 01001100 11001100 11001100 11001100 11001100 11001100 11001101
 π    10000010 01001001 00001111 11011010 10100010 00100001 01101000 11000010

◎X68000　倍精度（IEEEフォーマット）
+1.0  00111111 11110000 00000000 00000000 00000000 00000000 00000000 00000000
+0.5  00111111 11100000 00000000 00000000 00000000 00000000 00000000 00000000
-0.5  10111111 11100000 00000000 00000000 00000000 00000000 00000000 00000000
+0.1  00111111 10111001 10011001 10011001 10011001 10011001 10011001 10011010
 π    01000000 00001001 00100001 11111011 01010100 01000100 00101101 00100110
```

た。

数値演算コプロセッサとか浮動小数点演算コプロセッサというのは，その名のとおり浮動小数点演算をハードで行うものだ。CPUの機能をごく自然に拡張するものといえ，インテル8086/80286/80386用の8087/80287/80387やモトローラ68020/68030用の68881/68882が有名だ。多くの機種（たとえば98）ではCPUのすぐ隣にソケットが用意され，ここに浮動小数点演算コプロセッサを差し込むことで実数演算速度を数～数十倍に速めることができる（当然，ソフト側でコプロに対応していなければならないが)。

さて，コプロセッサに対応するためには，数値の表現形式も最初からコプロと同じものを使っていたほうが楽だ。そうすれば，いざというときに相互変換などの余分な処理をしなくてすむだろう。上記のコプロセッサはすべてIEEE規格を採用しているから，自然とソフト側の内部フォーマットもIEEE規格に従ったものとなるわけだ。

このようにIEEE規格の普及は数値演算コプロセッサの普及にかなり助けられたといってよいだろう。実際，IEEE規格はインテル社の社内規格を基にして決められたそうで，IEEE規格制定前に発売されていた8087は規格を先に広めておく役割を果たしたといえそうだ。

ところで，X68000で使われている（ただしオプション）浮動小数点演算プロセッサはMC68881だ。上でも述べたようにMC68881は本来MC68020用のものだから，MC68000をMPUとするX68000では使えないように見える。が，68020と68881との間の連絡手続きをエミュレートすることで68000でも68881を使うことができる。この場合，68881は純粋なコプロセッサではなく，どちらかといえば外部デバイスのような扱いとなる。実際X68000用の68881はソケットに差し込むといった簡単なものではなく，拡張スロットに収めるボードの形式で供給されている。

変換による誤差

ここまでの話はなんの役に立つのかと，いぶかしむ読者もいるだろう。内部表現なんてBASICなどの高級言語を使う分には関係ないじゃないかと思っているかもしれない。しかし，高級言語を使って数値演算を行うときにこそ，内部表現の知識が要求されるのだ。

数値演算を行うとき，必ず気をつけなければならないのは「誤差」の問題だ。特にコンピュータで数値演算を行う場合，内部で2進浮動小数点表現が用いられていることからくる誤差が入り込むことがある。

第1に，我々が普段使っている10進数をコンピュータに入力し，10進数から2進数へ変換する過程で誤差が生じる場合がある。たとえば，0.1を2進数で表してみると

.0110011001100……B

と無限に続く循環小数になってしまう。内部表現では仮数部のビット数に限りがあるので，途中でまるめなければならない。通常，格納できる範囲のすぐ下の桁を0捨1入し，その時点でどうしても誤差が入り込むことになる。

0.1は10回足せば1になるのは自明だが，このように誤差が入り込んだ0.1は10回足しても「1に近い数」にしかならない。信じられない人は次のようなプログラムで実験してみるとよい。なにやら汚い数字が並ぶことだろう。

```
10 A=0.1:B=0
20 FOR I=1 TO 10
30     B=B+A
40 NEXT
50 PRINT B-1.0
```

上のように結果を表示するだけならば，「あぁ，誤差が入ったんだな」と納得することもできるかもしれない。が，条件判断がからむと事態はかなりヤバくなる。50行目が，

```
50 IF B=1.0 THEN 100
```

となっていたらどういうことになるか，考えてみてほしい。この場合，正しく分岐させるためには，多少みっともなくても，

```
50 IF ABS(B-1.0)<1E-8
       THEN 100
```

のように，「Bと1.0の差が無視できるほど小さければ分岐する」という処理をしないといけない。実際，数値演算によく使われるFORTRANの世界ではこのような記述法は常識になっている。

第2に，結果を表示するときに行われる内部表現から10進数への変換の過程でも誤差が生じることがある。BASICでは実数を表示するときは10進で何桁と切って，その下の桁を四捨五入して表示している。たとえば，MZ-2500では単精度は10進8桁，倍精度は10進16桁で表示する。つまり，内部表現ではさらに細かい情報が含まれていたとしても，表示の段階でまるめられてしまうことになる。試しに上のプログラムの50行目を，

```
50 PRINT B
```

として実行してみてほしい。Bは1だと表示されるだろう。Bから1を引いたものは0にならなかったのにBは1だという不思議な結果が得られる。これも内部表現と10進数との相互変換による誤差のいたずらだ。

いまの例では，2進数への変換の際の誤差を表示するときに補ってくれたわけだから，ありがたいことともいえる。が，正しい値が表示されたとしても，内部での値はやはり誤差を含んでいるわけだから，この値を引き続き演算に使用すれば，誤差が引き継がれていくことになる。

また，シャープのマシンの話ではないが，マイクロソフトのBASICではかなり長い間（もしかするといまでも）10進への変換時の四捨五入にバグがあったという恐ろしい話もある。コンピュータの出力結果を過信してはならないという，マイクロソフトからの教訓だろうか。

演算による誤差

演算時においても，さまざまな理由で誤差が入り込んだり，精度が下がったり，情報が失われたりする場合が考えられる。

たとえば，

(A+B)−A

はBになるはずだが，Aに比べてBがずっと小さい場合は，正しい結果が得られないことがある。次のプログラムを実行してみてほしい。

```
10 A!=1234567.8
20 B!=.12345678
30 PRINT (A!+B!)-A!
```

MZ-2500で試してみたところ，結果は，

.12353516

となった。小数点以下4桁目以降が正しくないことがわかる。これは浮動小数点数の加算のアルゴリズムに原因がある。

話を簡単にするために10進数での浮動小数点表現で考えてみる。仮数部は3桁とし，123と1.23を足す様子を見てみよう。

123と1.23は浮動小数点表現に直すとそれぞれ，

$123 = .123 \times 10^3$

$1.23 = .123 \times 10^1$

となる。このままでは足せないので，人間が筆算でやるように，2数の小数点の位置を揃える（指数部を等しくする）必要がある。すると，

$.123 \quad \times 10^3$

$.00123 \times 10^3$

のようになるはずだが，仮数部は3桁だから入りきらない部分をなんとかしなければ

ならない。そこで4桁目で四捨五入して，

$.123\times10^3$

$.001\times10^3$

としてから足すことになり，

$.124\times10^3$

という結果が得られる。1.23の「0.23」の情報がなくなっているわけだ。

このように大きさ（指数部）が極端に違う2数の加減算では小さいほうの数の情報が失われることがある。もうひとつプログラム例を示そう。

```
10 A!=0
20 FOR I=1 TO 1000
30      Z!=I/1000!
40      A!=A!+Z!
50 NEXT
60 D#=A!:PRINT D#
```

このプログラムは0.001〜1までを0.001きざみで足すプログラムで，単精度で計算しておき，最後で倍精度の変数に代入することで，結果は倍精度で表示している。結果は500.5になり，単精度で計算しても十分正しい答えが得られることがわかる。

なお，FORループを，

```
20 FOR I=0.001 TO 1.0 STEP 0.001
```

のようにしていないのは，すでに述べたように0.001を2進数に変換した際の誤差が累積しないようにするためだ（実際に試してみるとよい）。

次に，このプログラムを少し変形して，大きいほうから足すようにしてみよう。

```
10 A!=0
20 FOR I=1 TO 1000
30      Z!=(1000-I)/1000!
40      A!=A!+Z!
50 NEXT
60 D#=A!:PRINT D#
```

演算の順序を変えただけなのに，今度は正しい答えよりいくぶん小さな結果が表示されるだろう。どこかで情報が落ちているからだ。なぜ，このような結果になるのかも考えてみてほしい。

また，上で示したプログラムにおいて，途中の演算もすべて倍精度で行うようにすれば，どちらでも正しい答えが表示される。仮数部のビット数が十分あるので，（あまり）情報が失われずにすんでいるためだ。まさに「倍精度」といえる。

最後に

コンピュータにおける数値演算はなかなか面白いテーマだ。ここで挙げた以外にも関数演算のアルゴリズム（EXP(−X)と1.0/EXP(X)が等しくならないのはなぜかとか）や浮動小数点表現数の四則演算のアルゴリズム，もう少し「高級」なところでは，多項式による関数近似の方法なども紹介したかったのだが，本稿では「誤差に気をつけようね」という話で押さえておいたほうが無難のようだ。最後に，ごく単純な誤差がらみの話をして終えることにする。

いくつかの例で，倍精度演算は単精度に比べてずっと誤差が少ないことは確認できた。まともな数値演算をするとなれば，多少時間がかかろうとやはり倍精度で行うのがよさそうだ。

しかし，勘違いから，倍精度演算の利点を生かせない場合がある。たとえば，なるべく高い精度で1/3を倍精度の変数に代入したいときに，

```
10 D#=1/3
```

としてしまってはいけない。一般にBASICでは黙っていれば単精度で演算が行われるので，いくら倍精度の変数に代入したところで，1/3は単精度で計算されてしまう。

こういった理由で，倍精度で結果を得たければ，

```
10 D#=1#/3#
```

のように，倍精度で演算することを明記しておかなければならない。

同じ理由で，

```
10 D#=SIN(0.5)
```

ではなく，

```
10 D#=SIN(0.5#)
```

を使うようにしたい。関数の演算は指定しなくても倍精度で行ってくれればよいのだが，多くの場合（少なくともMZ-2500では）特に指定しなければ結果は単精度で返ってくるようだ。

以上のように高い精度で演算したければ，倍精度の変数を使うだけで満足してしまわずに，定数データにも気をつかってもらいたい。

と，こんなところだ。少しは数値演算というものに興味を持っていただけただろうか。ここで述べたことが，いくらかでも読者の参考になれば幸いだ。また，数値演算を本格的にやろうという人は，10進1桁，いや1ビットにさえこだわって，より高精度な演算を目指してほしいと思う。先はまだ長い。

リスト1　実数の内部表現を調べるプログラム X1/MZ-2500

```
 10 ' 浮動小数点表現を調べるプログラム  MZ-2500/X1
 20 input D#
 30 'd#=pai(1#)
 40 F!=D#
 50 for I=1 to 5
 60     print right$("0000000"+bin$(asc(mid$(mks$(F!),I,1))),8);" ";
 70 next
 80 print
 90 for I=1 to 8
100     print right$("0000000"+bin$(asc(mid$(mkd$(D#),I,1))),8);" ";
110 next
120 print
130 goto 20
```

リスト2　実数の内部表現を調べるプログラム X68000

```
 1: /*
 2: **        浮動小数点表現を調べるプログラム  X68000
 3: */
 4:
 5: #include        <stdio.h>
 6: #include        <math.h>
 7:
 8: void main( argc, argv )
 9: int argc;
10: char **argv;
11: {
12:         char c;
13:         int i,j;
14:         double d;
15:         char *pnt;
16:
17:         if ( argc < 2 ) {
18:                 fprintf( stderr, "no arg¥n" );
19:                 exit( 1 );
20:         }
21:         while ( --argc ) {
22:                 d = atof( *++argv );
23: /*              d = PI;                 */
24:                 pnt = ( char * ) &d;
25:                 for ( i = 0; i < 8; i++ ) {
26:                         c = *pnt++;
27:                         for ( j = 0; j < 8; j++ ) {
28:                                 putchar( '0' + ( ( c >> 8 ) & 1 ) );
29:                                 c *= 2;
30:                         }
31:                         putchar( ' ' );
32:                 }
33:                 putchar( '¥n' );
34:         }
35: }
```

連立方程式は行列でいこう

Sohma Hidetomo
相馬 英智

方程式を解くことは，与えられた条件式から解の候補者を見つけ出し最後には真の解を言い当てようとする探偵小説のような面白さがあります。ここでは行列の概念を導入しながら，簡単なBASICを使って連立1次方程式を解いてみましょう。

突然だが，数学を考えたい

パソコンを使っているからといっても，誰もが数学が得意ではなかったりするのです。特に式をたくさん書いてごしょごしょ計算するのは私も不得手なわけで，そういった話はできれば遠慮したいなというのが正直なところです。

しかし，コンピュータの勉強を続けていくと，どうしても数学の知識が必要となります。といっても，広範な数学のすべてを理解しておく必要はありません。かなり以前から，コンピュータのためには数学の理解が重要だということはいわれていましたが，最近は，数学の多くの分野のなかでも離散数学といった分野を学習しておけばよいのではないかといわれています。この分野は他の数学の分野とはかなり独立していて，あまり数学に詳しくなくても学習することができるものと思われます。それでも，高校で習う程度の数学はどうしたって必要ですが。コンピュータを相手に仕事をしようと思っている人は考えておいたほうがよいでしょう。

さて，コンピュータと数学は切っても切れない関係にあります。もともとコンピュータは数学の問題を速く正確に解こうと生まれてきたものです。ご存じの方も多いと思いますが，最初のコンピュータは大砲の弾道を計算するために作られました。そしてコンピュータは数学の進歩を加速し，今やコンピュータを利用した新しい数学の分野が生まれつつあります。そこで，ここでは純粋な数学と，コンピュータのための数学などの違いについての話をしたいと思います。なにをやるかというと，数学の基本に返ったつもりで方程式，特に連立方程式というものに取り組んでみたいと思います。

方程式の基礎

方程式とは，まだわからない解を含んだ式のことです。このわからない値を未知数といい，xとかyなどとして表現します。たとえば，以下のような方程式が与えられたとします。

$x+3=5$

するとこの式はxは未知数で何かわからない値だけど，xが上の式を満たす値となることが表現されています。では具体的に，xの値はいくつになるのでしょうか。このように未知数の値を実際に得ることを，方程式を解くといい，得られた値を方程式の解といいます。上の式の場合，方程式を解くと，

$x=2$

となり，これが解です。

ここでさらに方程式というものを考えるために次の問題を考えてみましょう。

〈問題1〉

次の方程式をxについて解きなさい。

$ax=b$

あれ，未知数がたくさんあるぞと思った方がいるかもしれませんね。方程式をxについて解くということはxだけを未知数として，あとのaやbなどは何か決まった値として計算しなさいということです。具体的には$x=$なになにといった形の解を導き出すことになります。

さて，〈問題1〉は解けますか？　中学生以上の方なら答えはおわかりでしょう。そう，単に$x=b/a$なんて答えてはいけませんよね。解は以下のようになります。

〈問題1の解〉

$a\neq 0$のとき　$x=b/a$
$a=0$，$b=0$のとき　xはすべての数
$a=0$，$b\neq 0$のとき　xは解なし

どうして単純に$x=b/a$と答えてはいけないかというと，$a=0$のときには，解は$x=b/a$というかたちでは表現できないからです（0では割ることができない）。

この〈問題1〉では，式を方程式としているから数学の問題なのですが，これが仮に算数の問題だったとすると$x=b/a$と答えてもさほどおかしくありません。というのは，算数ではその式を満たすxの値を見つけることができればよいからです。

しかし数学の方程式の場合，与えられた式を満たす「すべてのx」の値を求めなければなりません。またその解法も直感的なものではなく，理論的に妥当で十分なものでなければならないのです。したがって数学というかぎりは，〈問題1の解〉のようなものにならなければいけないのです。

方程式を解くということは，厳密にいえば与えられた式を満たすすべての解を求めることで，だいたいの単純な方程式が解ける程度には解法の技術が高められています。では，解けない方程式があるかというと，実は結構単純なものでも解けない場合もあり，微分方程式と呼ばれる方程式では解けない場合のほうが多かったりします。

それでは，方程式を解く方法とはどのようなものでしょうか。これは解く方程式にそれぞれ独特の解く方法があります。しかし，基本は変わりません。それは以下の等号の定義と等式に関する4つの定理です。方程式には必ず等号“＝”が存在しますので，この定理は必ず使用可能です（ただし，これだけで必ず解が得られるとは限りません）。

まず等号の定義から見てみましょう。

0)　2つの式がそれぞれ唯一の値をもち，その値が等しいとき，2つの式は等号“＝”で結ぶことができる。

うーん，結構当たり前のことをいっているような気がしますね。では残る4つの定理を見てみましょう。

1)　等式の両辺に同じ値を加えても等式は成立する。

2)　等式の両辺から同じ値を差し引いても等式は成立する。

3)　等式の両辺に同じ値を掛けても等式は成立する。

4)　等式の両辺を0以外の同じ値で割っても等式は成立する。

さて，両辺という言葉が出てきましたが，これは等号の右の式を右辺，左の式を左辺といい両方いっしょに両辺という数学屋さんの言葉です。上の定理はいわれてみればそうかなといったものです。ただ注意して

ほしいのは定理の4)です。数学ではいかなる値であろうとも，その値を0で割ることは禁じられています。というよりは，0で割ったときの値が定義されていないのです。したがって値がわからないので，式を等号で結ぶことはできません。

では，これらを用いて問題を解いてみましょう。まずは最初に現れた方程式 $x+3=5$ です。方程式の解は $x=$ なになにのかたちになればいいので，そうなるように式を変形します。もうおわかりですね。両辺から3を引けばよいのです。すると，この方程式の解である $x=2$ が得られたではありませんか。

いまの例のように求める式と違う部分を式の他の辺に寄せ集めることで解が得られます。したがって，式中の各部分(厳密には各項)が移っていくように見えるので，このことを移項といいます。しかし実際は，方程式を解くときはこの定理をいちいち頭に浮かべる必要はありません。単純な方程式の場合，慣れてくれば反射的に解けるようになります。

そこで再び〈問題1〉を解いてみましょう。簡単簡単，まず両辺を a で割ります。すると $x=$ なになにのかっこうになりますね。おや，それでいいのでしょうか。a の値はわからないのです。したがってひょっとすると0かもしれないということですね。ということは，簡単に両辺を a で割ることはできません。そこで〈問題1の解〉を見ればわかるように a が0か，そうでないかを分けて答えているのです。

まず a が0でないと仮定しましょう。すると，両辺を a で割ることができて $x=b/a$ となります。

では $a=0$ のときは，どうなるのでしょうか。$a=0$ のとき，方程式の左辺 ax は x がいかなる値でも0となります(当たり前ですね)。そこで右辺 b が0であれば，この式は x がどんな値でも成立することになります。

逆に，b が0でなかったら，式は x の値に関わらず絶対に成立しません。したがって，“解がない”というのが答えとなります。解を求めよというのに“解がない”というのはちょっとおかしいような気がしますが，このへんが数学の正直なところで“ないものはない”と言ってしまっていいのです。

ともかく，これで〈問題1の解〉が得られました。ちなみに数学の試験では解答は，最初の〈問題1の解〉のようにしっかりと答えてください。数学の解答用紙には一定のフォーマットがあるので，それに従って書かないといけないようです。

先ほどの解を求める過程で気づいた方もいると思いますが，移項を行うことで，解は機械的に何も考えずに計算できます。というより，正しい解をできるだけ簡単に求めるように努力した結果，こうすれば必ず解が得られるのだという方法をあみ出したのです。与えられた手続きどおりに計算をする……これって，コンピュータの一番得意とするところじゃないですか。でも，こんな簡単な問題だと何もコンピュータにやらせても，あんまり意味がないので問題をもう少し難しくしましょう。

連立1次方程式を考える

それでは，次のような問題が与えられたとします。

〈問題2〉

以下の連立方程式を解きなさい。

$x+2y=5$

$5x+y=7$

さて，問題には2つの式があります。そして，やはり x と y という2つの未知数があります。一般に連立方程式とは複数の式と未知数を持っています。そして，各式の中で現れる同じ名前の未知数は同一のものです(当たり前かな？)。したがって，連立方程式の未知数は与えられた連立方程式を，すべて満足しなければなりません(すべての式が全部いっしょに成り立たなければなりません)。こうなってくると結構難しそうですね。

一般に方程式の中に現れる未知数を1元，2元と数えます。そしてもうひとつ，式中に現れる未知数の次元といったものがあります。これはそれぞれの未知数の最大の乗数を表現し，1次，2次と数えます。たとえば，〈問題1〉で与えられた式 $ax=b$ の場合，x の1乗があるので1次方程式，$x^2+x=3$ は2次方程式となります。したがって，〈問題2〉は2元1次連立方程式となります。今回は，この1次連立方程式について詳しくやっていこうという趣向ですので，他の方程式はそれなりの本などで勉強してくださいね。

では，〈問題2〉の場合どのようにしたら解が得られるのでしょうか。これも最初の方程式のように解くのですが，ちょっとした技術を使います。この計算をやってみたのが図1です。要するに，2つの式をうまいところ合わせて，x だけ，y だけの式を作ろうというわけです。x だけ，y だけの式となれば簡単に解けますよね。

このように複雑な問題は，できるだけ簡単な問題に分解したり，変換したりできればしめたものです。しかしコンピュータに解かせるのは，まだ早すぎます。というのは，私たちは，もっと問題について詳しく知る必要があります。なぜならコンピュータに問題を解かせるためには，確実に答えが得られる方法を知る必要がありますし(図1の方法で必ず解が導けるとは限らないから)，できれば一番楽な方法で解を得たいというのが人情です。そこで連立方程式の特性についてもっと詳しく取り組んでみましょう。

解が確実に得られる条件

まず，どんな場合も解が得られるのでしょうか。これって，意外と難しかったりするのです。そして，機械的に解くためには非常に重要なことです。つまり解法が完璧なものでないと，コンピュータによって得られた解が常に正しいとはいえず，意味のない計算をしてしまったことになるからです。

もちろん，どんな場合も計算できなければならないかというと，そうでもありません。こういう場合は解が得られないということがわかれば，その場合の計算はできないという結果を返すことで，得られた解の正当性が保てるようになります。したがって，計算をする前にちゃんとした解が得られるかどうかということを確認する必要があります。そこで以下の問題の場合を例に考えてみましょう。

〈問題3－0〉

以下の方程式を解きなさい。

$3x+y+2z=14$ ………(式1)

図1　連立方程式の解法(2元1次)

$$\begin{cases} x+2y=5 & —① \\ 5x+y=7 & —② \end{cases}$$

式②の両辺を2倍する。

$10x+2y=14$ —②′

式②′から式①の両辺をそれぞれ引く

$$\begin{array}{rr} & 10x+2y=14 \\ -) & x+2y=5 \\ \hline & 9x=9 \\ & x=1 \end{array}$$

式①の両辺を5倍する。

$5x+10y=25$ —①′

式①′から式②の両辺をそれぞれ引く

$$\begin{array}{rr} & 5x+10y=25 \\ -) & 5x+y=7 \\ \hline & 9y=18 \\ & y=2 \end{array}$$

答　$x=1$，$y=2$

実はこの問題の場合，基本的にこれ以上問題を簡単にするための変形をすることはできません。本当かなあと思う人は，試してみてください。このように未知数が3つもあるのに式が1つしかない場合，その式以上に簡単に解を限定することはできません。つまり単に（式1）を満たすx，y，zが解だよとしかいえないわけです。では，次の場合はどうでしょう。

〈問題3-1〉

以下の連立方程式を解きなさい。

$3x+y+2z=14$ ………（式1）

$x+5y-z=6$ ………（式2）

この場合（式1）と（式2）からはxとy，yとz，zとxといった関係が導き出されますが，そこまでで，解を1つに限定はできません。実際，〈問題3-1〉から得られた式をまとめたのが以下の関係式です。

$$60(x-3)=-154(y-1)=55(z-2)$$

そこで上の式値をtと置くと，解となるx，y，zは以下のように表せます。

$$x=t/60+3$$
$$y=-t/154+1$$
$$z=t/55+2$$

このtを特に媒介変数（パラメータ）と呼んでいます。したがって，tの値を決めると解となるx，y，zの値のうち1つが得られます。しかし，ここまで解を限定するのが限界です。これでも，〈問題3-0〉よりは解の限定ができていますが，1つに絞ることはできません。では，問題の式を3つにしてみましょう。

〈問題3-2〉

以下の連立方程式を解きなさい。

$3x+y+2z=14$ ………（式1）

$x+5y-z=6$ ………（式2）

$-2x-2y+4z=0$ ………（式3）

この場合は解を唯一のものとすることができます。図1に示した要領で計算したものを図2に記します。今度はうまくいきましたね。このように一般には連立方程式の場合，未知数の数だけの式が必要です。ただし，未知数の数だけあっても求まらない場合があります。その代表的な例が次の問題です。

〈問題3-3〉

以下の連立方程式を解きなさい。

$3x+y+2z=14$ ………（式1）

$x+5y-z=6$ ………（式2）

$6x+2y+4z=28$ ………（式4）

この場合，解は1つに限定できません。よく見てみると，（式4）は（式1）の両辺を2倍したもので，（式1）と同じことしか表現していません。したがって〈問題3-3〉は式は3本あっても，2本分しかないのと同じです。

それで解が1つに定まらないわけです。このように2つの式が実は同じ意味しか持たなかったり，ある式が他の式から導き出されたりする場合，その式は解を限定するのに役に立ちません。そこで数学では，ある式が他の式から導き出されないようになっているとき，各式は独立であるなどといい区別しています。このことから以下のようにいうことができます。

1次連立方程式の解を1つに定めるためには，独立した条件式が最低でも未知数の数だけ必要である。

では式が4つ以上あったらどうかなと思ったでしょう。条件式が多すぎると条件がきつくなり過ぎるために，解の限定が進みすぎてしまい条件を満たす解が存在しなくなります。とはいえ，4つ以上の式があってもそのうち独立してないものがあって，実質的に3本の式と同じようになる場合は，解を1つに定めることが可能です。これで解が得られる条件は，はっきりしました。連立方程式の解を導き出すのに必要な条件は，いままでの話でわかってもらえたと思います。あとは，連立方程式を計算機内でどのように表現するかということです。

図2　連立方程式の解法(3元1次)

$$\begin{cases} 3x+y+2z=14 & \text{—①} \\ x+5y-z=6 & \text{—②} \\ -2x-2y+4z=0 & \text{—③} \end{cases}$$

①+2×②

$3x+y+2z=14$
$+)\ 2x+10y-2z=12$
$5x+11y=26$ —④

4×②+③

$4x+20y-4z=24$
$+)\ -2x-2y+4z=0$
$2x+18y=24$ —⑤

2×④−5×⑤

$10x+22y=52$
$-)\ 10x+90y=120$
$-68y=-68$
$y=1$

同様にして

答　$x=3$，$y=1$，$z=2$

行列と連立方程式

ここからは与えられた連立1次方程式をどのように解くかではなく，どのようにうまく表現するかということについて考えていきたいと思います。このことは，データ構造に関する分野にかなり近いものと思われます。ただ実際にはこのような数値演算の場合，かなり単純なデータ構造である配列でけっこう間に合うことが多いのです。これは数学自体が極めて美しく整理されていて，まとまっているからです。そのため科学技術演算用（数値演算用）のFORTRANや，それから派生したBASICなどの言語は，配列一筋の頑固者になっているのでした。

話を数学に戻しましょう。実際，連立1次方程式は行列と呼ばれるかたちで表現することができ，これをうまく利用して行列のかたちで処理する（解く）という体系が数学では完全に作られています。では，行列とはどのようなものでしょうか。行列とは複数の値を（要素といいます）ひとまとめにして1つの値とするものです。下に例をあげましょう。

$$\begin{pmatrix} 3 & 1 & 2 \\ 5 & 8 & 10 \\ 2 & 1 & 1 \end{pmatrix}$$

このように行列とは，縦と横に値を並べたものです。そして横に値が並んでいるのを行といい，上から第1行，第2行と呼びます。同様に縦の並びを列といい，前から第1列，第2列と呼びます。つまり行列とは行と列という2つの値の並びのことで，ほとんど漫才コンビの名前みたいな命名法が取られたことがわかります。数学って，このへんが割りといい加減なんですよね。

そしてm行n列の構成の行列を$m\times n$の行列といいます。この行列にもさまざまなものが考えられ，名前がついているものもあります。まずは$n\times n$の行列，これを正方行列といいます。また，行または列が1つしかない行列には，ベクトルなんてしゃれた名前がついています。

そして大事なことはこの行列にも“＋－×”といった計算が行えることです。おっとその前に，行列における等号“＝”の意味を知っておく必要があります。つまり，どういう場合に2つの行列の式が，等しいといえるのかということです。これを正確に知らずして足し算などは語れません。数学ではこれを“行列における等号の定義”なんて難しそうにいいますが，私たちは数学屋さんではないので堅苦しい話は抜きにします。

行列において等しいということは，次の条件を満たせば十分です。つまり2つの行列が，同じ大きさ（同じ行数で同じ列数）で，同じ行同じ列の要素がすべて等しければ，等号が成立します。したがって行列内のすべての要素について，それと同じ位置

にある要素が等しくなければならないのです（言われてみると，当然という気もするなあ）。

では，足し算（和）と引き算（差）を見てみましょう（例1）。これは同じ行と列の値どうしを加える（引く）ことで結果の行列が得られます。

皆さんはもうお気づきだと思いますが，この行列の足し算，引き算は，同じ大きさ（同じ行で，同じ列）の行列どうしでしか行うことはできません。では掛け算（積）はどうでしょうか。これは，ちょっと複雑な計算となります。まずは例2を見てもらいましょう。

よーく見るとわかると思いますが（わかるかなー），前の行列の各行と後ろの行列の各列のそれぞれの要素を掛けて和をとっています。行列の掛け算は計算は複雑ですが，その分よく用いられます。

ここで注意してほしいのは，行列の積が計算できる条件です。和や差の場合は，前の行列と後ろの行列の行どうし，列どうしが同じ大きさでないとできませんでした。しかし積の場合，前の行列の行の大きさと，後ろの行列の列の大きさが等しいことが計算できる条件です。これはとても重要なことです。

さて，この行列の積で〈問題3－2〉の問題を表現してみましょう。すると以下のようになります。

$$\begin{pmatrix} 3 & 1 & 2 \\ 1 & 5 & -1 \\ -2 & -2 & 4 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 14 \\ 6 \\ 0 \end{pmatrix}$$

どうです。きれいなものでしょう。この積の部分を計算して等号の条件から各要素を展開すると，元の式になることがわかると思います。

ここで，未知数の入った行列（ベクトル）に注目してください。この問題を解くにあたって，私たちが未知数について知っておくことは，どの未知数が何を意味しているかということだけです。したがって未知数の名前である x，y，z はあくまでも未知数間の区別をするためにつけた名前なのです。もし式の形などでその区別が表現されれば，x，y，z などといった情報は不要となります。さて，そんなにうまく，いくものでしょうか。

行列と掃き出し法

先ほどの〈問題3－3〉を，行列で表現したものをよく見てください。未知数のベクトルと積をとっている行列（未知数の係数行列といいます）の第1列は，すべて未知数 x の係数となっています。おっと係数というのは式の中で，その未知数や変数などに掛けてある数のことです。同様に第2列，第3列は，それぞれ y，z の係数となっていますね。

このように，行列に連立1次方程式を書き直したときに，行列の構造が各係数の情報を表現できるようになっているのでした。こうしてみると，行列のような構造を持ったデータが持つ意味の重要性がわかると思います。

ついでに先ほどの行列とベクトルの積の右辺の部分を含めて，以下のような行列を作ります。

$$\begin{pmatrix} 3 & 1 & 2 & 14 \\ 1 & 5 & -1 & 6 \\ -2 & -2 & 4 & 0 \end{pmatrix}$$

この場合も最後の列が，定数であることを区別できます。もうお気づきの方も多いことと思いますが，このように行列の形に連立方程式を格納しようとしているのは，コンピュータ（特にBASICやFORTRAN）の得意な配列をうまく使おうというためです。素人でも，行列と配列がかなり近い関係があるのがわかると思います。これについては，詳しく説明しなくても大丈夫ですよね。

それではいよいよ，どのようにして連立方程式を解くのかを考えてみましょう。最も基本的な方法は，図1や図2でやった方法です。これを簡単な手続きとして，まとめたものが“掃き出し法”です。名前がちょっと汚いような気がしますが，極めて基本的で有名な方法（アルゴリズム）です。

数学の言葉で行列を掃き出すということは，行列の意味を保ったままどこかを基準（要）として，ある行や列の要素を0になるように変形することです。掃き出し法では〈問題3－3〉の場合，以下のような形に掃き出そうとします。

$$\begin{pmatrix} 1 & 0 & 0 & 3 \\ 0 & 1 & 0 & 1 \\ 0 & 0 & 1 & 2 \end{pmatrix}$$

これを未知数を含んだ行列の形に直すと，以下のようになります。

$$\begin{pmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{pmatrix} \begin{pmatrix} x \\ y \\ z \end{pmatrix} = \begin{pmatrix} 3 \\ 1 \\ 2 \end{pmatrix}$$

つまり，この形は求めるべき連立方程式の解を示しています。ただし注意しなければならないのは，変形を施すことで行列の値は変わっているので等号では結べません。

さて，どのようにして変形すればよいのでしょうか。具体的には，プログラムを見てもらったほうがてっとり早いと思います。掃き出し法で3元1次連立方程式を解くプログラムを，リスト1に示します。まあ，読んでみてください。このやり方は以下のような処理をします。

まず第1行第1列の要素を取り出し，この部分が1になるように第1行の要素をそれぞれ割ります。次に第1行第1列を除く第1列の要素が0になるように，行列の要素どうしの引き算を行います。このようにすることで，第1列は目的の行列と同じになります。これを各列に前から順番に施すことで，最後には目的の行列が得られることとなるでしょう。これが掃き出し法です。

プログラムを見ればわかるように，かなり単純なプログラムです。でも実はプログラムには2つの欠点があります。ひとつは未知数の数が違う連立方程式に対応できるように拡張できないことです。これはBASIC

例1　行列の和と差

$$\begin{pmatrix} 3 & 1 & 2 \\ 5 & 8 & 10 \\ 2 & 1 & 1 \end{pmatrix} + \begin{pmatrix} 0 & 1 & 3 \\ 1 & 2 & 1 \\ 5 & 7 & 4 \end{pmatrix}$$

$$= \begin{pmatrix} 3+0 & 1+1 & 2+3 \\ 5+1 & 8+2 & 10+1 \\ 2+5 & 1+7 & 1+4 \end{pmatrix}$$

$$= \begin{pmatrix} 3 & 2 & 5 \\ 6 & 10 & 11 \\ 7 & 8 & 5 \end{pmatrix}$$

$$\begin{pmatrix} 3 & 1 & 2 \\ 5 & 8 & 10 \\ 2 & 1 & 1 \end{pmatrix} - \begin{pmatrix} 1 & 8 & 1 \\ 2 & 6 & 3 \\ 7 & 12 & 0 \end{pmatrix}$$

$$= \begin{pmatrix} 3-1 & 1-8 & 2-1 \\ 5-2 & 8-6 & 10-3 \\ 2-7 & 1-12 & 1-0 \end{pmatrix}$$

$$= \begin{pmatrix} 2 & -7 & 1 \\ 3 & 2 & 7 \\ -5 & -11 & 1 \end{pmatrix}$$

例2　行列の積

$$\begin{pmatrix} 3 & 1 & 2 \\ 5 & 8 & 10 \\ 2 & 1 & 1 \end{pmatrix} \begin{pmatrix} 0 & 1 & 3 \\ 1 & 2 & 1 \\ 5 & 7 & 4 \end{pmatrix}$$

$$= \begin{pmatrix} 3\times0+1\times1+2\times5 & 3\times1+1\times2+2\times7 & 3\times3+1\times1+2\times4 \\ 5\times0+8\times1+10\times5 & 5\times1+8\times2+10\times7 & 5\times3+8\times1+10\times4 \\ 2\times0+1\times1+1\times5 & 2\times1+1\times2+1\times7 & 2\times3+1\times1+1\times4 \end{pmatrix}$$

$$= \begin{pmatrix} 11 & 19 & 18 \\ 58 & 91 & 63 \\ 6 & 11 & 11 \end{pmatrix}$$

が再帰的プログラミングができないということに起因しています。でもX-BASICでは再帰ができるので拡張ができるでしょう。X68000ユーザーの人は各自がんばってくださいね。

そしてもうひとつの欠点は，このプログラムを使っているとわかりますが，たまに（かなりひんぱんに）計算できないことがあります。これは，与えた式が解を1つに限定するのに不十分なとき（これは先ほどお話ししましたね）と，計算中に要素を1にしたい位置なのにたまたまそこが0となっている場合です。

これはアルゴリズム自体の欠点なのですが，式を入力する順番を変えることなどで解決できます。しかし，できればここまでコンピュータにさせたいものです。これは結構簡単に解決できます。問題は1にしたいn行n列の値が0になっているときは，それ以下の行でn列が値が0でない行と取り換えればいいのです。これは簡単な改造ですみますので，やっぱりがんばってください。とにかく大事なことは，数学の技術をどのようにコンピュータのプログラミングに組み入れて使うかということなのですから。

さて，掃き出し法は単純でプログラミングには持ってこいです。しかし数学的にはもっと美しい解法があります。それがCramerの公式です。

美しいCramer公式と縁起ものの行列式

Cramer法とはどんなやり方でしょうか。それを話す前に，このCramer法の方針といったものについて話しましょう。

このCramer法は，連立方程式が与えられたら，各系数を代入するだけで解が得られるような公式を作ってしまおうといったものです。これならばコンピュータは先ほどの掃き出し法のように値を確認しながら処理を行うといったものではなく，直接与えられた係数をもとに一気に計算するだけでよくなります。また数学的には解を直接，式で表現できるようになり，とても美しくなります。

では，Cramerの公式を図3に示します。どうです，美しいでしょう。ただ式中に行列のようだけどちょっと違うものがありますね。これが行列式というものです。この行列式は結構複雑なのであまり触れたくはないのですが，これがわからないとCramerの公式はわかりませんので簡単に説明します。

リスト1-a　掃き出し法による解法(X68000)

```
100 /*
110 /*  掃き出し法による3元1次連立方程式の解の算出
120 /*
130 /*                              Oh!X  88_8
140 /*
150 int i,j
160 float value
170 dim float x(2,3)
180 /*
190 /*  式値の入力
200 /*
210 cls
220 for i=0 to 2
230         print i+1;"本目の式を入力します。"
240         for j=0 to 2
250                 print "X(";j+1;")=";
260                 input " ",value
270                 x(i,j)=value
280         next
290         input "式値 = ",value
300         x(i,3)=value
310 next
320 /*
330 /*  入力された式の出力
340 /*
350 for i=0 to 2
360         for j=0 to 2
370                 print x(i,j);"*X(";j+1;")";
380                 if j<>2 then print " + ";
390         next
400         print " = ";x(i,3)
410 next
420 /*
430 /*  掃き出し法の実行
440 /*
450 Gauss()
460 /*
470 /*  結果の出力
480 /*
490 for i=0 to 2
500         print "X(";i+1;") = ";x(i,3)
510 next
520 end
530 /*
540 /*  掃き出し法
550 /*
560 func Gauss()
570 int i,j,k
580         for i=0 to 2
590                 for j=i+1 to 3
600                         x(i,j)=x(i,j)/x(i,i)
610                 next
620                 for j=0 to 2
630                         if i<>j then {
640                                 for k=i+1 to 3
650                                         x(j,k)=x(j,k)-x(j,i)*x(i,k)
660                                 next
670                         }
680                 next
690         next
700 endfunc
```

リスト1-b　掃き出し法による解法(X1/X1turbo)

```
100 REM 掃き出し法による3元1次連立方程式の解の算出
110 REM
120 REM                          Oh!X  88_8
130 REM
140 DIM x(2,3)
150 REM
160 REM 式値の入力
170 REM
180 CLS
190 FOR i=0 TO 2
200     PRINT i+1;"本目の式を入力します。"
210     FOR j=0 TO 2
220             PRINT "X(";j+1;")=";
230             INPUT " ",va
240             x(i,j)=va
250     NEXT j
260     INPUT "式値 = ",va
270     x(i,3)=va
280 NEXT i
290 REM
300 REM 入力された式の出力
310 REM
320 FOR i=0 TO 2
330     FOR j=0 TO 2
340             PRINT x(i,j);"*X(";j+1;")";
350             IF j<>2 THEN PRINT " + ";
360     NEXT j
370     PRINT " = ";x(i,3)
380 NEXT i
390 REM
400 REM 掃き出し法の実行
410 REM
420 GOSUB "掃き出し法"
430 REM
440 REM 結果の出力
450 REM
460 FOR i=0 TO 2
470 PRINT "X(";i+1;") = ";x(i,3)
480 NEXT i
490 END
500 REM
510 LABEL "掃き出し法"
520 REM
540 FOR i=0 TO 2
550     FOR j=i+1 TO 3
560             x(i,j)=x(i,j)/x(i,i)
570     NEXT j
580     FOR j=0 TO 2
590             IF i=j GOTO "対角要素"
600             FOR k=i+1 TO 3
610                     x(j,k)=x(j,k)-x(j,i)*x(i,k)
620             NEXT k
630             LABEL "対角要素"
640     NEXT j
650 NEXT i
660 RETURN
```

行列式は見た目はかなり行列に似ていますが，両者には大きな違いがあります。それは行列式は普通の値を持つということです。普通の値とは3とか2.64とかです。このような値を“スカラー”といっています。いうまでもなく，行列って行列の値しか持ちませんよね。図4に行列式の例をあげます。行列式は，

$$\begin{vmatrix} 6 & 1 & 8 \\ 7 & 5 & 3 \\ 2 & 9 & 4 \end{vmatrix}$$

のように書き表します。

そして，この行列式の値は360です。このように行列式は行列とたいへん似ていますが，スカラー値を持っているわけですから，等号は普通の意味で使えます。

さて，どのようにして行列式の値を求めるのでしょうか。行列式の値は，その行列式の要素から計算されます。しかし実はこれが，たいへん面倒臭いのです。そして，これこそコンピュータにやらせるべき仕事です。なぜ面倒臭いかというと，行列式の大きさ（行数と列数）が変わると行列式の値を求める式も変わってくるからです（とはいえ，ちゃんと規則性はあるんだよ）。

では，これを計算するプログラムを組んでみましょうと言いたいとこなんだけど，今回は連立方程式がテーマだし……といってさぼってしまうのでした。やっぱり行列や行列式は，それなりのところで勉強したほうがいいでしょう奥が深いですからね。でも，あっさりあきらめてしまうのは残念なので，簡単な行列式の場合の値の計算の

図3　Cramerの公式

連立1次方程式

$$\begin{cases} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n = b_1 \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n = b_2 \\ \quad\vdots \\ a_{n1}x_1 + a_{n2}x_2 + \cdots + a_{nn}x_n = b_n \end{cases}$$

のとき，（aは係数，xは未知数）

$$x_i = \frac{\begin{vmatrix} a_{11} & a_{12} & \cdots & b_1 & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & b_2 & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & b_n & \cdots & a_{nn} \end{vmatrix}}{\begin{vmatrix} a_{11} & a_{12} & \cdots & a_{1n} \\ a_{21} & a_{22} & \cdots & a_{2n} \\ \vdots & \vdots & & \vdots \\ a_{n1} & a_{n2} & \cdots & a_{nn} \end{vmatrix}}$$

第i列をbで置き換える。

リスト2-a　Cramerの公式による解法(X68000)

```
100 /*
110 /*  Cramerの公式による3元1次連立方程式の解の算出
120 /*
130 /*                              Oh!X  88_8
140 /*
150 int i,j
160 float value
170 dim float x(2,3)
180 /*
190 /*  式値の入力
200 /*
210 cls
220 for i=0 to 2
230      print i+1;"本目の式を入力します。"
240      for j=0 to 2
250           print "X(";j+1;")=";
260           input " ",value
270           x(i,j)=value
280      next
290      input "式値 = ",value
300      x(i,3)=value
310 next
320 /*
330 /*  入力された式の出力
340 /*
350 for i=0 to 2
360      for j=0 to 2
370           print x(i,j);"*X(";j+1;")";
380           if j<>2 then print " + ";
390      next
400      print " = ";x(i,3)
410 next
420 /*
430 /*  Cramerの公式による解の算出と出力
440 /*
450 d=det(-1)
460 for i=0 to 2
470      print "X(";i+1;") = ";det(i)/d
480 next
490 end
500 func float det(m;int)
510 int i,j,k,ans
520 dim float y(2,3)
530 for i=0 to 2
540      for j=0 to 2
550           if j=m then {
560                y(i,j)=x(i,3)
570           } else {
580                y(i,j)=x(i,j)
590           }
600      next
610 next
620 ans=y(0,0)*y(1,1)*y(2,2)+y(0,1)*y(1,2)*y(2,0)+y(0,2)
*y(1,0)*y(2,1)-y(0,2)*y(1,1)*y(2,0)-y(0,1)*y(1,0)*y(2,2)
-y(0,0)*y(1,2)*y(2,1)
630 return(ans)
640 endfunc
```

リスト2-b　Cramerの公式による解法(X1/X1turbo)

```
100 REM Cramerの公式による3元1次連立方程式の解の算出
110 REM
120 REM                             Oh!X 88_8
130 REM
140 DIM x(2,3),y(2,3)
150 REM
160 REM 式値の入力
170 REM
180 CLS
190 FOR i=0 TO 2
200      PRINT i+1;"本目の式を入力します。"
210      FOR j=0 TO 2
220           PRINT "X(";j+1;")=";
230           INPUT " ",va
240           x(i,j)=va
250      NEXT j
260      INPUT "式値 = ",va
270      x(i,3)=va
280 NEXT i
290 REM
300 REM 入力された式の出力
310 REM
320 FOR i=0 TO 2
330      FOR j=0 TO 2
340           PRINT x(i,j);"*X(";j+1;")";
350           IF j<>2 THEN PRINT " + ";
360      NEXT j
370      PRINT " = ";x(i,3)
380 NEXT i
390 REM
400 REM Cramerの公式による解の算出と出力
410 REM
420 i=-1
430 GOSUB "Cramerの公式"
440 div=ans
450 FOR i=0 TO 2
460      GOSUB "Cramerの公式"
470      PRINT "X(";i+1;") = ";ans/div
480 NEXT i
490 END
500 REM
510 LABEL "Cramerの公式"
520 REM
530 FOR j=0 TO 2
540      FOR k=0 TO 2
550       IF i=k THEN y(j,k)=x(j,3) ELSE y(j,k)=x(j,k)
560      NEXT k
570 NEXT
580 ans=y(0,0)*y(1,1)*y(2,2)+y(0,1)*y(1,2)*y(2,0)+y(0,2)
*y(1,0)*y(2,1)-y(0,2)*y(1,1)*y(2,0)-y(0,1)*y(1,0)*y(2,2)
-y(0,0)*y(1,2)*y(2,1)
590 RETURN
```

仕方を図4に示します。

というわけで，連立方程式に戻りましょう。このCramerの公式を用いたプログラムをリスト2に示します。どうです，あんまり美しくないでしょう。数学的に美しいCramerの公式も，プログラムにするとかたなしです。

ここで考えてほしいのは，数学的な美しさとプログラム的な美しさとの違いです。数学は実利をまったく無視しているわけではないのですが結構実利を離れて，抽象的で思想的な美しさをもつ分野です。これに対し，コンピュータというのは実利重視で，どろどろとしたものをたくさん引きずったものです。この両者の違いはあまりにも大きく，多くの問題を引き起こします。では，コンピュータにとって最も望ましい解き方（アルゴリズム）とは，どのようなものでしょうか。

三角分解による連立方程式の解法

実際にコンピュータに連立方程式を解かせようとするとき，いちばん気をつけなければならないことは何でしょうか。それは，おそらく計算の効率だと思います。パソコンで面白半分にプログラムを組んでいるときはどうでもいいことですが，ちゃんと満足のできる，ましてや他人に見せるプログラムならばなおさらです。また実際にコンピュータに連立方程式を解かせようとするときがどんな場合かと考えれば，解くべき式がかなり膨大なときであることが予想されます。

この場合，掃き出し法やCramerの公式などは処理の効率が悪く，かなり使いづらいものとなります。では，効率を上げるためにはどうすればいいのでしょうか。効率を上げるためのひとつの基準となるのは処理速度です。そこで話を処理速度に絞って，それを上げるためにはどうすればよいのかということを，考えてみたいと思います。

処理速度を上げるためには，いくつかの手段が考えられます。大事なことは要領よく処理をするということです。これは効率のよいアルゴリズムを採用するといったことにつながります。たとえば多くのデータのソートだったら有効なアルゴリズムとしてはクイックソートが有名です。

しかし，このアルゴリズムの選択をするときに気をつけなければならないことがあります。それは処理すべきデータの数やデータの特性，処理を行うときのさまざまな制約条件によって，どのアルゴリズムが有効かということが大きく左右されるということです。そのため，ひとつの処理をとってもたくさんのアルゴリズムが存在しています。投入されたデータによって処理の戦略を変えるというダイナミックプログラミングという手法さえあるぐらいです。

さてそこで，最終的にこの連立方程式の解法でとったアルゴリズムは何かというと，三角分解（LU分解）を用いたアルゴリズムです。なぜこのアルゴリズムを取ったかというと，以下のような理由があります。

第1に単純な連立方程式ならコンピュータにいちいち入力するほうが時間がかかってしまうのではないかということ。第2に，とにかく掛け算が少ないことです。マシン語を知っている人はわかると思いますが，掛け算は足し算より数倍時間がかかるのです。その代わり少しループが増えていますが掛け算の回数もかなり少なくなっていて，たぶん，こちらのほうが速いと思います。このプログラムをリスト3に示します。

三角分解とはなんだろうと思った方も多いでしょう。これは通常の行列を，以下のような2つの行列の積に分解してしまおうというものです。

$$\begin{pmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{pmatrix} = \begin{pmatrix} l_{11} & 0 & 0 \\ l_{21} & l_{21} & 0 \\ l_{31} & l_{32} & l_{33} \end{pmatrix} \begin{pmatrix} u_{11} & u_{12} & u_{13} \\ 0 & u_{22} & u_{23} \\ 0 & 0 & u_{33} \end{pmatrix}$$

例：

$$\begin{pmatrix} 8 & 4 & 12 \\ 2 & 11 & 13 \\ 14 & 13 & 31 \end{pmatrix} = \begin{pmatrix} 4 & 0 & 0 \\ 1 & 5 & 0 \\ 7 & 3 & 1 \end{pmatrix} \begin{pmatrix} 2 & 1 & 3 \\ 0 & 2 & 2 \\ 0 & 0 & 4 \end{pmatrix}$$

こうすることで連立1次式の1つの未知数の値が確定してしまいます。あとはこれを代入して，残りの解を単純に計算するだけです。結構，要領よくやっているでしょう。なお三角分解については，やっぱりそれなりのところで勉強したほうがいいと思います。

とにかくコンピュータで良好なアルゴリズムを選ぶと，足し算より掛け算は時間がかかるとか，メモリに限界があるとか，数学では考えられない恐ろしく泥臭いことが起きるのです。そしてこれらを考慮してプログラムを組むことが，効率のよい処理をしようとすると要求されるのです。

これは数学の見地からいえば，とんでもないことです。したがって数学的な（論理的な）美しさなんてものは，コンピュータにとってはあまり意味のないことです。一般に数学とコンピュータとはかなり仲がいいと思われているのですが，本当だろうかと考え込んでしまう私なのでした。

最後に

なんだかんだといっても，コンピュータと数学はお互いにたいへん影響を与えあっています。昔から数学のさまざまなアイデアがコンピュータに取り込まれてきましたし，最近数学のなかにもコンピュータの優れた計算力を背景とした（要するにコンピュータに莫大な計算をさせて行う）分野が現れてきました。しかし数学でいう演算とコンピュータでいう演算では大きな意味の開きがあることが，今回の話でわかっていただけたと思います。

でも，コンピュータを詳しく知りたかったら数学は不可欠です。はっきり言って私は数学が苦手です。特に式を書いてごしょごしょするのは本当に苦手です。でも，コンピュータを使っているうちにやっぱり数学は必要だということで，私だってそこそこに勉強しているのですよ。

図4　行列式

$$|a| = a$$

$$\begin{vmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{vmatrix} = a_{11}a_{22} - a_{12}a_{21}$$

$$\begin{vmatrix} a_{11} & a_{12} & a_{13} \\ a_{21} & a_{22} & a_{23} \\ a_{31} & a_{32} & a_{33} \end{vmatrix} = a_{11}a_{22}a_{23} + a_{12}a_{23}a_{31} + a_{21}a_{32}a_{13} - a_{31}a_{22}a_{13} - a_{11}a_{32}a_{23} - a_{21}a_{12}a_{33}$$

$$\vdots$$

リスト3-a　三角分解を用いた解法（X68000）

```
100 /*
110 /*   三角分解を用いた3元1次連立方程式の解の算出
120 /*
130 /*                  Oh!X  88_8
140 /*
```

リスト3-b　三角分解を用いた解法（X1/X1turbo）

```
100 REM 三角分解を用いた3元1次連立方程式の解の算出
110 REM
120 REM                         Oh!X 88_8
130 REM
140 DIM x(2,3),L(2,2),U(2,2),y(2)
```

```
150 int i,j
160 float value
170 dim float x(2,3),L(2,2),U(2,2)
180 /*
190 /*  式値の入力
200 /*
210 cls
220 for i=0 to 2
230       print i+1;"本目の式を入力します。"
240       for j=0 to 2
250             print "X(";j+1;")=";
260             input " ",value
270             x(i,j)=value
280       next
290       input "式値 = ",value
300       x(i,3)=value
310 next
320 /*
330 /*  入力された式の出力
340 /*
350 for i=0 to 2
360       for j=0 to 2
370             print x(i,j);"*X(";j+1;")";
380             if j<>2 then print " + ";
390       next
400       print " = ";x(i,3)
410 next
420 /*
430 /*  三角分解の実行
440 /*
450 Decomp()
460 /*
470 /*  結果の算出
480 /*
490 Solve()
500 /*
510 /*  結果の出力
520 /*
530 for i=0 to 2
540       print "X(";i+1;") = ";x(i,3)
550 next
560 end
570 /*
580 /*  三角分解
590 /*
600 func Decomp()
610 int i,j,k
620 float s
630       for i=0 to 2
640             L(i,0)=x(i,0)
650       next
660       for i=1 to 2
670             U(0,i)=x(0,i)/L(0,0)
680       next
690       for i=1 to 2
700             for j=i to 2
710                   s=0
720                   for k=0 to i-1
730                         s=L(j,k)*U(k,i)+s
740                   next
750                   L(j,i)=x(j,i)-s
760             next
770             for j=i+1 to 2
780                   s=0
790                   for k=0 to i-1
800                         s=L(i,k)*U(k,j)+s
810                   next
820                   U(i,j)=(x(i,j)-s)/L(i,i)
830             next
840       next
850       for i=0 to 2
860             U(i,i)=1
870       next
880 endfunc
890 /*
900 /*  結果の計算
910 /*
920 func Solve()
930 int i,j,k
940 float s
950 dim float y(2)
960       y(0)=x(0,3)/L(0,0)
970       for i=1 to 2
980             s=0
990             for j=0 to i-1
1000                  s=L(i,j)*y(j)+s
1010            next
1020            y(i)=(x(i,3)-s)/L(i,i)
1030      next
1040      x(2,3)=y(2)
1050      for i=0 to 1
1055            j=1-i
1060            s=0
1070            for k=j+1 to 2
1080                  s=U(j,k)*x(k,3)+s
1090            next
1100            x(j,3)=y(j)-s
1110      next
1120 endfunc
```

```
150 REM
160 REM 式値の入力
170 REM
180 CLS
190 FOR i=0 TO 2
200       PRINT i+1;"本目の式を入力します。"
210       FOR j=0 TO 2
220             PRINT "X(";j+1;")=";
230             INPUT " ",va
240             x(i,j)=va
250       NEXT j
260       INPUT "式値 = ",va
270       x(i,3)=va
280 NEXT i
290 REM
300 REM 入力された式の出力
310 REM
320 FOR i=0 TO 2
330       FOR j=0 TO 2
340             PRINT x(i,j);"*X(";j+1;")";
350             IF j<>2 THEN PRINT " + ";
360       NEXT j
370       PRINT " = ";x(i,3)
380 NEXT i
390 REM
400 REM 三角分解の実行
410 REM
420 GOSUB "三角分解"
430 REM
440 REM 結果の算出
450 REM
460 GOSUB "算出"
470 REM
480 REM 結果の出力
490 REM
500 FOR i=0 TO 2
510       PRINT "X(";i+1;") = ";x(i,3)
520 NEXT i
530 END
540 REM
550 LABEL "三角分解"
560 REM
570 FOR i=0 TO 2
580       L(i,0)=x(i,0)
590 NEXT i
600 FOR i=1 TO 2
610       U(0,i)=x(0,i)/L(0,0)
620 NEXT i
630 FOR i=1 TO 2
640       FOR j=i TO 2
650             s=0
660             FOR k=0 TO i-1
670                   s=L(j,k)*U(k,i)+s
680             NEXT
690             L(j,i)=x(j,i)-s
700       NEXT j
710       FOR j=i+1 TO 2
720             s=0
730             FOR k=0 TO i-1
740                   s=L(i,k)*U(k,j)+s
750             NEXT k
760             U(i,j)=(x(i,j)-s)/L(i,i)
770       NEXT j
780 NEXT i
790 FOR i=0 TO 2
800       U(i,i)=1
810 NEXT i
820 RETURN
830 REM
840 LABEL "算出"
850 REM
860 y(0)=x(0,3)/L(0,0)
870 FOR i=1 TO 2
880       s=0
890       FOR j=0 TO i-1
900             s=L(i,j)*y(j)+s
910       NEXT j
920       y(i)=(x(i,3)-s)/L(i,i)
930 NEXT i
940 x(2,3)=y(2)
950 FOR i=0 TO 1
960       j=1-i
970       s=0
980       FOR k=j+1 TO 2
990             s=U(j,k)*x(k,3)+s
1000      NEXT k
1010      x(j,3)=y(j)-s
1020 NEXT i
1030 RETURN
```

特集　真夏の夜の数値演算

iがあるからむずかしい

Mukouhara Ayumu
向原 あゆむ

「iってよくわからないけど……，苦しむ感じが素敵」というわけで，とってもとっても難しい複素数のお話です。ここでは，複素数に関する基礎知識と，BASICで簡単に複素数の演算を行うためのサブルーチン集を紹介しましょう。

方程式から生まれたi

皆さんは，虚数というものを知っていますか。これは自分自身を掛け合わせる（2乗する）と負の値になるという数のことです。たとえば，

$$x^2=-1 \qquad \cdots\cdots\cdots(1)$$

という方程式を考えてみましょう。通常の数（実数）なら2乗すると0か正の値になりますから，この方程式の解は存在しません。そこで，2乗すると負になるような数を「無理やり考える」のです。たとえば，

$$i^2=-1 \qquad \cdots\cdots\cdots(2)$$

であるような数iを考えます。すなわち，

$$i=\sqrt{-1}$$

というわけです。これが数学や物理学でちょくちょくお目にかかるiという数のもっとも単純な定義になります。iという名前は「imaginary number（想像上の数）」の頭文字を取ったものです。

これでさっきの方程式(1)の解はもちろんiということになります。また，このiに関して$-i$（iの-1倍）という数を考えると，

$$\begin{aligned}(-i)^2&=(-i)\times(-i)\\&=(-1)\times i\times(-1)\times i\\&=(-1)\times(-1)\times i\times i\\&=1\times i\times i\\&=i^2=-1\end{aligned}$$

となりますから，$-i$も方程式(1)の解であることがわかります。ただしこの変換はそれほど当たり前というわけではなく，

- $a\times b=b\times a$
- $(a\times b)\times c=a\times(b\times c)$

という関係が成立することを仮定しています。この関係は実数同士では明らかですが，虚数においても成立すると仮定するわけですね。まさに仮定だらけの「空虚」な数ですが，iを導入したことにより，方程式(1)の解を2つ（iと$-i$）求めることができました。次に，

$$x^2=-n^2 \qquad \cdots\cdots\cdots(3)$$

（nは0以上の実数）

という方程式を考えましょう。この方程式の解も虚数になりますが，この方程式の解はiを使って$n\times i$と$-n\times i$と表すことができます(わかりますよね)。このようにすべての虚数はiの定数倍として表すことができるためiを虚数単位と呼びます。

それでは，もっと一般的な2次方程式，

$$ax^2+bx+c=0\ (a, b, c\text{は実数}) \qquad \cdots\cdots(4)$$

について考えます。この方程式の解は判別式を，

$$D=b^2-4ac$$

とするとき，

$$x=(-b\pm\sqrt{D})/2 \qquad \cdots\cdots\cdots(5)$$

となります。もし，Dが0以上なら(5)は2つの実数になりますが，Dが負のときは実数解が存在しません。Dが負のときの方程式(4)の解は，虚数単位iを使って

$$x=(-b\pm i\sqrt{-D})/2 \qquad \cdots\cdots\cdots(6)$$

と表せます。実数の範囲内だけで2次方程式の解を求めようとすると，解が存在しない場合と存在する場合に分かれてしまいますが，虚数iの助けを借りると必ず2つの解を求めることができるようになります。逆の見方をすれば，2次方程式に必ず2つの解を持たせるためにiという虚数が考えられたのだということもできます。

ところで，(5)と(6)の式を見てわかるとおり，2次方程式の2つの解は必ず，

$$a+b\cdot i\ (a, b\text{は実数})$$

という形式をしています。そして，実数と虚数を組み合わせてできるこのような数を複素数（complex number：複合した数，決してコンプレックスを持った数という意味ではない）と呼んでいます。

さて，2次方程式に2つの解を持たせるために虚数iというものが考えられたといいましたが，同様にして3次方程式，4次方程式，……に次数と同じ個数の解を持たせるためにはどんな数を考えなければならないのでしょうか。実はそれらの高次方程式を解く場合にも複素数の範囲，つまり実数と虚数の組み合わせだけで十分であることがわかっています。これが「代数学の基本定理」というやつです。すなわち，「n次方程式は複素数の範囲でn個の解を持つ」ということです。

それでは，n次方程式の解がn個求まるようになると何かいいことがあるのでしょうか。実はなんの役にも立ちません。しかし，同じ複素数が物理学の分野では物体の運動を記述する場合に重要な役割を果たしていることを知っておきましょう。

そこでは，iという数値は-1の平方根という形式的な意味だけでなく，もっと現実的な意味を持っているのです。それについてはあとで説明しますが，それにしても，虚構の世界から発想された複素数が現実の世界と関わりを持っているとは驚きでもありますね。もっとも，「負の数」は実際の世界には存在しませんが数学や物理学で重要な役割を果たしていることを考えると，「虚数」や「複素数」をそれほど特別視する必要もないのですけどね。

複素数の演算

ここでは，複素数の基本的な演算について押さえておきましょう。虚構の世界で扱う場合も現実の世界で扱う場合も複素数としての演算は同じですから，これがすべての基本になります。

複素数は，

$$z=x+yi \qquad \cdots\cdots\cdots(7)$$

（x，yは実数，iは虚数単位）

のかたちで表される数です。このときxを実数部，yを虚数部と呼びます。2つの複素数$a+bi$と$c+di$が等しいためには$a=c$かつ$b=d$でなくてはなりません。すなわち，複素数の実数部と虚数部は「独立」で，実数部と虚数部を定めると複素数は一意に決まります。また，複素数には四則演算が定義されます。これは以下のような規則になります。

加算

$$\begin{aligned}&(a+bi)+(c+di)\\&=(a+c)+(b+d)i\end{aligned} \qquad \cdots\cdots\cdots(8)$$

減算

$$(a+bi)-(c+di)$$

$$= (a-c) + (b-d)i \quad \cdots\cdots\cdots(9)$$

乗算

$$(a+bi)\times(c+di)$$
$$= ac + adi + bci - bd$$
$$= (ac-bd) + (bc+ad)i \quad \cdots\cdots\cdots(10)$$

除算

$$\frac{a+bi}{c+di}$$
$$= \frac{(a+bi)(c-di)}{(c+di)(c-di)}$$
$$= \frac{ac-adi+bci+bd}{c^2+d^2}$$
$$= \frac{(ac+bd)-(ad-bc)i}{c^2+d^2} \quad \cdots\cdots\cdots(11)$$

どれも実数の演算と同じ式の変形をしながら，途中で出てくるi^2を－1に置き換えているだけですから，これは自然な定義といえるでしょう。そして，ここで大事なことは実数の演算で使っていた法則がそのまま使えるということです。実際，複素数の虚数部を0とするとそれは実数そのものですね。複素数とは実数を拡張した普遍的な数ということができるのです。

複素数とベクトル

複素数は実数と虚数の組み合わせですが，もっと現実的な意味を持たせることができます。先に述べたように，複素数は実数部と虚数部の値を決定すると一意に定まります。そこで，

$$a+bi$$

という複素数を，

$$(a, b)$$

と記述する方法が考案されました。この記述ではiという虚数単位は表立って出てきませんが，その表す内容は同じことです。

さて，この表記を眺めれば，これがxy平面上の1点の表記によく似ていることがわかります。たとえば，(a, b)という表記はxy平面上においてx座標がaでy座標がbである点の表記と同一です。つまり，x軸が実数で，y軸が虚数であるような座標系(平面)を考えると，すべての複素数はその平面内の1点として認識することができるようになります(図1)。

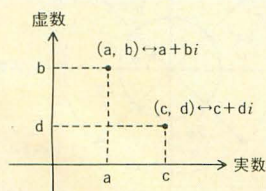
図1　複素数と複素平面

ここにおいて，話はちょっと飛躍しますが，a＋biという複素数はxy平面上の位置ベクトル（原点を始点とする）と同一視できるようになります。ベクトルというと矢印(有向線分）を思い浮かべるかもしれませんが，高校や大学の数学では物体の位置を示す1点のことを言います（その点とは原点を始点とする有向線分の終点に他ならないのですが)。

図2　複素数の四則演算

実際のところ，複素数の演算はベクトルの演算と同じ演算が成立します。たとえば，式(8)，(9)に対応して，

加算

$$(a, b) + (c, d)$$
$$= (a+c, b+d) \quad \cdots\cdots\cdots(12)$$

減算

$$(a, b) - (c, d)$$
$$= (a-c, b-d) \quad \cdots\cdots\cdots(13)$$

という関係が成立します。これはベクトルの加減算と同一ですね。

しかし，式(10)や(11)に見られるような乗除算はベクトルの演算であまり見かけることがないように思われます。それでは，いったいなんだろうかということで思い出されるのは行列とベクトルの積です。いま，

$$(a, b)\times(1, 0) = (a, b)$$
$$(a, b)\times(0, 1) = (-b, a)$$

であることを考えると，

$$(a, b)\times(c, d)$$

という演算は(c, d)というベクトルに行列

$$\begin{pmatrix} a & -b \\ b & a \end{pmatrix}$$

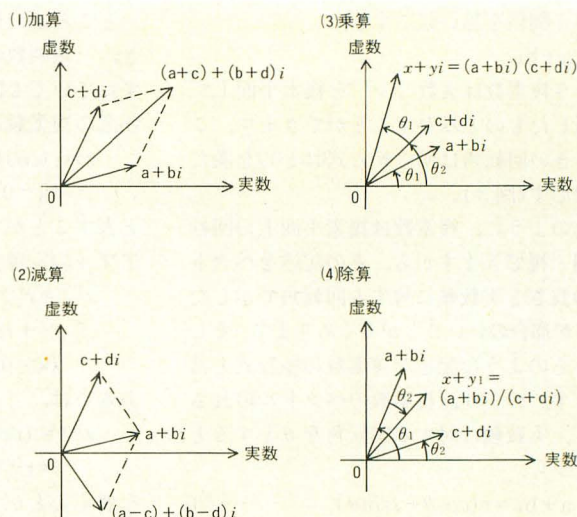
を掛けることに等しくなります（1次変換を勉強したことのある人ならわかりますよね)。さらに，

$$\cos\theta = \frac{a}{\sqrt{a^2+b^2}} \quad \cdots\cdots\cdots(14)$$

$$\sin\theta = \frac{b}{\sqrt{a^2+b^2}} \quad \cdots\cdots\cdots(15)$$

であるようなθを考えると上の行列は，

$$\sqrt{a^2+b^2}\begin{pmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{pmatrix} \quad \cdots\cdots\cdots(16)$$

と変形できます。これはベクトルを角度θだけ回転させて，$\sqrt{a^2+b^2}$倍する行列に他なりません（これも1次変換で出てきましたね)。すなわち複素数の掛け算はベクトルの回転と拡大（あるいは縮小）を意味するのです。

それでは除算はどうでしょうか。ある複素数で割るということは逆数を掛けるということですから，除算も回転と拡大にほかなりません。実際，

$$\frac{1}{c+di}$$

は，

$$\frac{c-di}{(c+di)(c-di)}$$
$$= \frac{c}{c^2+d^2} - \frac{di}{c^2+d^2}$$

ですから乗算と同様に考えると，複素数の割り算は，

$$\frac{1}{\sqrt{c^2+d^2}}\begin{pmatrix} \cos(-\theta) & -\sin(-\theta) \\ \sin(-\theta) & \cos(-\theta) \end{pmatrix} \quad \cdots\cdots\cdots(17)$$

ただし，

$$\cos\theta = \frac{c}{\sqrt{c^2+d^2}} \quad \cdots\cdots\cdots(18)$$

$$\sin\theta = \frac{d}{\sqrt{c^2+d^2}} \quad \cdots\cdots\cdots(19)$$

という行列を掛けることが割り算ということになります。これは要するに逆方向の回転ですね。

どうです，おぼろげながら複素数の演算の意味が見えてきたと思いませんか。もっと意味をはっきりさせるために図2に複素数の四則演算のイメージを示しておきましょう。

複素数の極形式

複素数の乗算はベクトルの回転を意味していますが，複素数はそれ自体実数を回転させていたものと見ることができます。たとえば，先に挙げた，

$$(a,b)\times(1,0)=(a,b) \quad \cdots\cdots(20)$$

という関係を思い起こすと，

$$a+bi$$

という複素数は実数“1”を複素平面上で回転したものとみなすことができます。このときの回転角は明らかに式(14)と(15)を満たすθです(図3)。

このように，複素数は複素平面上の回転と同一視できますから，その記述をベクトルの長さと実数軸に対する回転角で示したほうが都合のいいことが多くあります。そして，そのような記述を複素数の極形式と言います。つまり，複素数のベクトルの長さをr，実数軸に対する回転角をθとすると，図3から，

$$a+bi=r(\cos\theta+i\sin\theta) \quad \cdots\cdots(21)$$

$$\text{ただし } r=\sqrt{a^2+b^2}$$

であることがわかります。

実数部と虚数部を決めれば複素数が一意に定まることを先に説明しましたが，その代わりにベクトルの長さと回転角（正式には偏角と言う）を決めても複素数を一意に定めることができます。このrとθの組は極座標と呼ばれます。この極座標(極形式)を用いれば，回転と絡めて複素数の乗除算に関する次の公式を直感的に理解することができます。つまり，

$$z_1=r_1(\cos\theta_1+i\sin\theta_1)$$

$$z_2=r_2(\cos\theta_2+i\sin\theta_2) \quad \cdots\cdots(22)$$

とするとき，

$$z_1z_2=r_1r_2(\cos(\theta_1+\theta_2)+i\sin(\theta_1+\theta_2)) \quad \cdots\cdots(23)$$

$$\frac{z_1}{z_2}=\frac{r_1}{r_2}(\cos(\theta_1-\theta_2)+i\sin(\theta_1-\theta_2)) \quad \cdots\cdots(24)$$

$$z^n=\{r(\cos\theta+i\sin\theta)\}^n$$

$$=r^n(\cos(n\theta)+i\sin(n\theta)) \quad \cdots\cdots(25)$$

が成立します。特に(25)は「ド・モアブルの定理」と呼ばれる有名な関係ですが，zを1回掛けることでθだけ回転が行われることを考えれば自明な関係ですね。

また，有名な「オイラーの公式」，

$$e^{i\theta}=\cos\theta+i\sin\theta \quad \cdots\cdots(26)$$

を用いれば(23)，(24)，(25)の関係は，複素数の回転を考えなくても，次のように純粋に指数法則からだけでも理解することもできます。

$$z_1z_2=r_1e^{i\theta_1}\cdot r_2e^{i\theta_2}$$

$$=r_1r_2e^{i\theta_1+i\theta_2}$$

$$=r_1r_2e^{i(\theta_1+\theta_2)} \quad \cdots\cdots(27)$$

$$\frac{z_1}{z_2}=\frac{r_1e^{i\theta_1}}{r_2e^{i\theta_2}}=\frac{r_1}{r_2}e^{i\theta_1-i\theta_2}$$

$$=\frac{r_1}{r_2}e^{i(\theta_1-\theta_2)} \quad \cdots\cdots(28)$$

$$z^n=(re^{i\theta})^n$$

$$=r^ne^{in\theta} \quad \cdots\cdots(29)$$

ところで，複素数を極形式で記述したとき，三角関数の性質から偏角は2πの整数倍ずれていても同じことになります。そこで，任意の複素数は，

$$z=r(\cos(\theta+2k\pi)+i\sin(\theta+2k\pi))$$

$$(k=0,1,2,3,\cdots\cdots) \quad \cdots\cdots(30)$$

と表すことができます。このとき，「ド・モアブルの定理」(25)から，

$$z^{1/n}=r^{1/n}\{\cos((\theta+2k\pi)/n)$$

$$+i\sin((\theta+2k\pi)/n)\}$$

$$(k=0,1,2,3,\cdots\cdots) \quad \cdots\cdots(31)$$

あるいは，

$$z^{1/n}=(re^i)^{1/n}$$

$$=r^{1/n}\cdot e^{i(\theta+2k\pi)/n} \quad \cdots\cdots(32)$$

を導くことができます。

この関係は偏角が$(\theta+2k\pi/n)$であるような複素数をn乗すると，偏角がθに戻るということを意味します。k=0,1,2,……，(n−1)のとき式(31)または(32)で示される複素数はすべて異なります（図4）から，これらの式は複素数のn乗根をすべて（代数学の基本定理からn個あるはず）求める式ということができます。極形式を使用すると複素数の乗除算（特にベキ乗とベキ乗根）を簡単に計算できるようになるのです。

余談ですが，式(26)の「オイラーの公式」でθをπとすると，

$$e^{i\pi}=\cos(\pi)+i\sin(\pi)$$

$$=-1$$

ですから，両辺の自然対数を取って，

$$i\pi=\log_e(-1) \quad \cdots\cdots(33)$$

という関係を得ることができます。この関係を使うと対数関数の定義域を正の数だけでなく負の数にも拡大することができます。複素数って偉大ですね。

物理学への応用

複素数は数学だけでなく物理学にも応用されていて，現実の物体の運動を解き明かすための助けとなっています。その典型は振動する物体の記述です。単純な例は単振動でしょう。単振動の変位は，

$$z=C\cdot e^{i\omega t} \quad \cdots\cdots(34)$$

という式の実数部で表されます。わざわざzを使うのは微分や積分が簡単にできるからです。言い換えれば微分・積分をすっきりとしたかたちで行うためなのです。単振動の速度は変位zを時間tで微分した，

$$\frac{dz}{dt}=C\cdot i\omega e^{i\omega t}$$

$$=i\omega\cdot z \quad \cdots\cdots(35)$$

の実数部で与えられますが，このとき微分という操作が，$i\omega$を掛けるという操作に置き換えられることに気づくでしょう。これがわかれば，単振動の加速度は速度に$i\omega$を掛けて（時間tで微分して），

$$\frac{d^2z}{dt^2}=(i\omega)\frac{dz}{dt}=(i\omega)(i\omega\cdot z)$$

$$=-\omega^2\cdot z \quad \cdots\cdots(36)$$

となります。これは，単振動の運動方程式（微分方程式）に他なりません。

実際，物理学において複素数は運動方程式（微分方程式）を解く場合に威力を発揮します。減衰振動の運動方程式，

$$m\frac{d^2x}{dt^2}+R\frac{dx}{dt}+kx=0 \quad \cdots\cdots(37)$$

を解いてみましょう。この方程式の解が，

$$x=C\cdot e^{\lambda t} \quad \cdots\cdots(38)$$

で与えられるとすると，

$$\frac{dx}{dt}=\lambda\cdot C\cdot e^{\lambda t}=\lambda x \quad \cdots\cdots(39)$$

$$\frac{d^2x}{dt^2}=\lambda\frac{dx}{dt}=\lambda^2x \quad \cdots\cdots(40)$$

ですから，(37)，(39)，(40)から，

$$(m\lambda^2+R\lambda+k)x=0 \quad \cdots\cdots(41)$$

が成立します。したがって，λの満たすべき関係は，

$$m\lambda^2+R\lambda+k=0 \quad \cdots\cdots(42)$$

となります。2次方程式の解は一般に2つの複素数ですから，減衰振動も式(34)と同様の式(38)で与えられることがわかりますね。何かキツネにつままれたような気がしないでもありませんが，これは運動方程式の正式な解き方のひとつなのです。

ところで，単振動において微分は$i\omega$を掛けることでした。それでは積分はどうでしょう。お察しのとおり，それは$i\omega$で割

図3 複素数の極形式

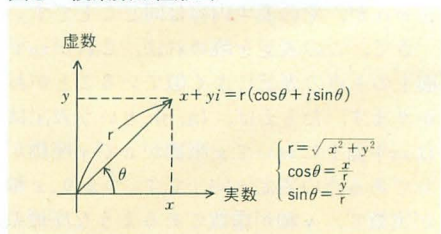
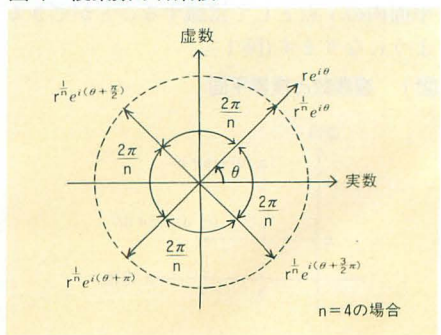



図4 複素数のn乗根

ることになります(積分は微分の逆演算)。これからピーンときた人がいるかもしれませんが，$i\omega$ というものは結局は工学の分野で多用されるラプラス変換（大学で習うことですからわからなくていいです）のパラメータsと同じようなものといえるでしょう。このアイデアまで行き着くと，複素数という虚構の数も現実と関わりの深い数であることがおぼろげながら見えてくるのです。

少しプログラムを

これまで複素数について説明してきたわけですが，このまま終わってしまってはこの雑誌が『大学への数学』とか『数学セミナー』といった類の雑誌と間違えられてしまうかもしれませんね。Oh!Xはパソコンの雑誌ですから，ちょいと簡単なプログラムを作ってみました。

それがリスト1(X-BASIC)およびリスト2(HuBASIC)です。どちらも内容は同じで，複素数の演算を行うためのサブルーチン集と実数係数の2次方程式と3次方程式の解を求めるためのプログラムです。BASIC上でサブルーチンを呼び出して遊んでみてください。少しは複素数が身近に感じられてくるでしょう（ということはないか)。

プログラムを実行すると変数zfr，zfi，zfpr，zfpt，および，配列RR，RIが定義されます。これは複素数演算の結果を格納するための変数と配列の宣言で，これが終わったらダイレクトモードでサブルーチンを呼び出すことができます。

ただし，リスト2のX1版では実行時に多少注意が必要です。それは，一般のBASICでは，サブルーチンに引数を渡すことができないためで，サブルーチンを呼ぶ前に引数と同じ名前の変数に値を入れてやらなければなりません。

プログラムで定義されているサブルーチンの簡単な説明を以下にまとめておきますが，zadd(xr, xi, yr, yi)と書かれているサブルーチンをHuBASICで実行するためには，

XR＝値：XI＝値：YR＝値：YI＝値：GOSUB“ZADD”

というぐあいにします（名前はすべて大文字にしてください)。

なお，RECTANGULAR_FORMというサブルーチンの引数RADIUSは，X1では予約語（RADで始まる）に引っ掛かるためXRADIUSとしてあります。

といったところで今回の複素数の話を終わることにいたしましょう。プログラムのなかでどのようなことが行われているかは，これまでの説明を思い出しながら読んでみてくださいね。

サブルーチンの概要

zadd（xr, xi, yr, yi）
　複素数（xr, xi）と（yr, yi）の和を計算して実数部を変数zfrに，虚数部を変数zfiに入れる。

zsub(xr, xi, yr, yi)
　複素数（xr, xi）と（yr, yi）の差を計算して実数部を変数zfrに，虚数部を変数zfiに入れる。

zmul(xr, xi, yr, yi)
　複素数（xr, xi）と（yr, yi）の積を計算して実数部を変数zfrに，虚数部を変数zfiに入れる。

zdiv(xr, xi, yr, yi)
　複素数（xr, xi）と（yr, yi）の商を計算して実数部を変数zfrに，虚数部を変数zfiに入れる。

polar-form(x, y)
　複素数（x, y）を極形式に変換して長さを変数zfprに，偏角を変数zfptに入れる。偏角は0～2πの範囲に入るように補正する。

rectangular_form(radius, theta)
　長さradius，偏角thetaで与えられる極形式の複素数を直交座標形式に変換して実数部を変数zfrに，虚数部を変数zfiに入れる。

polar_zmul(xr, xi, yr, yi)
　複素数（xr, xi）と（yr, yi）の積を計算して実数部を変数zfrに，虚数部を変数zfiに入れる。計算は極形式に変換してから行う。

polar_zdiv(xr, xi, yr, yi)
　複素数（xr, xi）と（yr, yi）の商を計算して実数部を変数zfrに，虚数部を変数zfiに入れる。計算は極形式に変換してから行う。

zpow(xr, xi, n)
　複素数（xr, xi）のn乗を計算して実数部を変数zfrに，虚数部を変数zfiに入れる。計算は極形式に変換してから行う。

znroot(xr, xi, n, ith)
　複素数（xr, xi）の$i$th番目のn乗根を計算して実数部を変数zfrに，虚数部を変数zfiに入れる。計算は極形式に変換してから行う。

dim2_equation(b, c)
　実数係数の2次方程式

$$x^2+bx+c=0$$

の解を求める。ひとつめの解の実数部をRR(0)に，虚数部をRI(0)に入れる。2つめの解の実数部をRR(1)に，虚数部をRI(1)に入れる。

dim3_equation(b, c, d)
　実数係数の3次方程式

$$x^3+bx^2+cx+d=0$$

の解を求める。ひとつめの解の実数部をRR(0)に，虚数部をRI(0)に入れる。2つめの解の実数部をRR(1)に，虚数部をRI(1)に入れる。3つめの解の実数部をRR(2)に，虚数部をRI(2)に入れる。

dim3_check(b, c, d)
　dim3_equationを実行して得られる3次方程式の解，RR(0)～RR(2)，RI(0)～RI(2)から，

$$x^3+bx^2+cx+d$$

の値を計算する。正しい解であれば値が0（計算誤差が入ってくるので実際は0に近い値）になるはず。

※3次方程式の解法(おまけ)

$$x^3+bx^2+cx+d=0 \quad \cdots\cdots(a)$$

で，

$$X=x+b/3 \quad \cdots\cdots(b)$$

と置くと，方程式(a)は，

$$X^3+pX+q=0 \quad \cdots\cdots(c)$$

のかたちに変形できる。
　一方，因数分解の公式，

$$Y^3-u^3-v^3-3uv$$
$$=(Y-u-v)(Y-\omega u-\omega^2v)(Y-\omega^2u-\omega v)$$

(ωは1でない1の3乗根の片方) ……(d)

から，

$$uv=-p/3 \quad \cdots\cdots(e)$$
$$u^3+v^3=-q \quad \cdots\cdots(f)$$

であるようなuとvの組を求めれば，Xは，

$$u+v$$
$$\omega u+\omega^2v$$
$$\omega^2u+\omega v$$

となる。このとき，(b)の関係により，Xの実数部からb/3を引けば方程式(a)の解が求まる。

(e)と(f)から，u^3とv^3は方程式，

$$Z^2+qZ-p^3/27=0 \quad \cdots\cdots(g)$$

の解になる（解と係数の関係)。これからu^3またはv^3が求まり，関係(e)を満たすようなuとvの組を求めることができる。このように，3次方程式は2次方程式に帰着して解くことができる。

リスト1　複素数演算のためのサブルーチン(X-BASIC)

```
   1 float zfr,zfi,zfpr,zfpt
   2 dim float RR(3),RI(3) : end
1000 /*
1010 /*         複素数の和
1020 /*
1030 func zadd(xr;float,xi;float,yr;float,yi;float)
1040   zfr=xr+yr        : zfi=xi+yi
1050 endfunc
1060 /*
1070 /*         複素数の差
1080 /*
1090 func zsub(xr;float,xi;float,yr;float,yi;float)
1100   zfr=xr-yr        : zfi=xi-yi
1110 endfunc
1120 /*
1130 /*         複素数の積
1140 /*
1150 func zmul(xr;float,xi;float,yr;float,yi;float)
1160   zfr=xr*yr-xi*yi : zfi=xr*yi+xi*yr
1170 endfunc
1180 /*
1190 /*         複素数の商
1200 /*
1210 func zdiv(xr;float,xi;float,yr;float,yi;float)
1220   float r
1230   r=yr*yr+yi*yi
1240   zfr=(xr*yr+xi*yi)/r : zfi=(-xr*yi+xi*yr)/r
1250 endfunc
1260 /*
1270 /*         極座標形式への変換
1280 /*
1290 func polar_form(x;float,y;float)
1300   zfpr=sqr(x*x+y*y)
1310   if x=0# and y>=0# then zfpt=pi()/2#  : return()
```

```
1320   if x=0# and y<0#  then zfpt=1.5#*pi(): return()
1330   zfpt=atan(y/x)
1340   if x>0# and y>=0# then return()
1350   if x>0# and y<0#  then zfpt=pi(2)+zfpt: return()
1360   zfpt=pi()+zfpt
1370 endfunc
1380 /*
1390 /*         直交座標形式への変換
1400 /*
1410 func rectangular_form(radius;float,theta;float)
1420   zfr=radius*cos(theta)
1430   zfi=radius*sin(theta)
1440 endfunc
1450 /*
1460 /*         複素数の積（極座標による）
1470 /*
1480 func polar_zmul(xr;float,xi;float,yr;float,yi;float)
1490   float r1,r2,t1,t2
1500   polar_form(xr,xi) : r1=zfpr : t1=zfpt
1510   polar_form(yr,yi) : r2=zfpr : t2=zfpt
1520   rectangular_form(r1*r2,t1+t2)
1530 endfunc
1540 /*
1550 /*         複素数の商（極座標による）
1560 /*
1570 func polar_zdiv(xr;float,xi;float,yr;float,yi;float)
1580   float r1,r2,t1,t2
1590   polar_form(xr,xi) : r1=zfpr : t1=zfpt
1600   polar_form(yr,yi) : r2=zfpr : t2=zfpt
1610   rectangular_form(r1/r2,t1-t2)
1620 endfunc
1630 /*
1640 /*         複素数のベキ乗（極座標による）
1650 /*
1660 func zpow(xr;float,xi;float,n;int)
1670   float r1,t1
1680   polar_form(xr,xi) : r1=zfpr : t1=zfpt
1690   rectangular_form(pow(r1,n),n*t1)
1700 endfunc
1710 /*
1720 /*         複素数のベキ乗根（極座標による）
1730 /*
1740 func znroot(xr;float,xi;float,n;int,ith;int)
1750   float r1,t1
1760   polar_form(xr,xi) : r1=zfpr : t1=zfpt
1770   rectangular_form(pow(r1,1#/n),(t1+pi(2)*ith)/n)
1780 endfunc
1790 /*
1800 /*         2次方程式の解法
1810 /*
1820 func dim2_equation(b;float,c;float)
1830   float x1,y1,x2,y2,d
1840   d=b*b-4#*c
1850   znroot(d,0#,2,0) : x1=zfr/2# : y1=zfi/2#
1860   znroot(d,0#,2,1) : x2=zfr/2# : y2=zfi/2#
1870   zadd(-b/2#,0#,x1,y1)
1880   RR(0)=zfr : RI(0)=zfi
1890   zadd(-b/2#,0#,x2,y2)
1900   RR(1)=zfr : RI(1)=zfi
1910 endfunc
1920 /*
1930 /*         3次方程式の解放
1940 /*
1950 func dim3_equation(b;float,c;float,d;float)
1960   float p,q,w1r,w1i,w2r,w2i,t1r,t1i
1970   float u3r,u3i,v3r,v3i,ur,ui,vr,vi
1980   p=c-b*b/3# : q=2#*b*b*b/27#-b*c/3#+d
1990   dim2_equation(q,-1#*p*p*p/27#)
2000   u3r=RR(0) : u3i=RI(0)
2010   v3r=RR(1) : v3i=RI(1)
2020   if (u3r=0) and (u3i=0) then {
2030     ur=0 : ui=0
2040     if (v3r=0) and (v3i=0) then {
2050        vr=0 : vi=0
2060     } else {
2070        znroot(v3r,v3i,3,0) : vr=zfr : vi=zfi
2080     }
2090   } else {
2100     znroot(u3r,u3i,3,0) : ur=zfr : ui=zfi
2110     zdiv(-1#*p/3#,0,ur,ui) : vr=zfr : vi=zfi
2120   }
2130   w1r= -0.5# : w1i=sqr(3)/2# : w2r=w1r : w2i= -w1i
2140   zadd(ur,ui,vr,vi)
2150   RR(0)=zfr-b/3# : RI(0)=zfi
2160   zmul(ur,ui,w1r,w1i) : t1r=zfr : t1i=zfi
2170   zmul(vr,vi,w2r,w2i)
2180   zadd(t1r,t1i,zfr,zfi)
2190   RR(1)=zfr-b/3# : RI(1)=zfi
2200   zmul(ur,ui,w2r,w2i) : t1r=zfr : t1i=zfi
2210   zmul(vr,vi,w1r,w1i)
2220   zadd(t1r,t1i,zfr,zfi)
2230   RR(2)=zfr-b/3# : RI(2)=zfi
2240 endfunc
2250 /*
2260 /*         検算プログラム（3次方程式）
2270 /*
2280 func dim3_check(b,c,d)
2290   float rr,ri
2300   int   i
2310   for i=0 to 2
2320      print "第";i+1; "の根:";RR(i);"+";RI(i);"i"
2330      zpow(RR(i),RI(i),3)
2340      rr=zfr : ri=zfi
2350      zmul(RR(i),RI(i),RR(i),RI(i))
2360      rr=rr+zfr*b : ri=ri+zfi*b
2370      rr=rr+RR(i)*c+d : ri=ri+RI(i)*c
2380      color 2
2390      print "方程式の値:";rr;"+";ri;"i"
2400      color 3
2410   next
2420 endfunc
```

リスト2　複素数演算のためのサブルーチン(HuBASIC)

```
1000 DEFDBL A-Z:DEFINT I:DIM RR(3),RI(3)
1010 END
1020 '
1030 '          複素数の和
1040 '
1050 LABEL"ZADD":ZFR=XR+YR:ZFI=XI+YI:RETURN
1060 '
1070 '          複素数の差
1080 '
1090 LABEL"ZSUB":ZFR=XR-YR:ZFI=XI-YI:RERURN
1100 '
1110 '          複素数の積
1120 '
1130 LABEL"ZMUL":ZFR=XR*YR-XI*YI:ZFI=XR*YI+XI*YR:RETURN
1140 '
1150 '          複素数の商
1160 '
1170 LABEL"ZDIV":ZFI=YR*YR+YI*YI:ZFR=(XR*YR+XI*YI)/ZFI
1180 ZFI=(-XR*YI+XI*YR)/ZFI:RETURN
1190 '
1200 '          極座標形式への変換
1210 '
1220 LABEL"POLAR_FORM":ZFPR=SQR(X*X+Y*Y)
1230 IF(X=0)AND(Y>=0)THEN ZFPT=PAI(.5):RETURN
1240 IF(X=0)AND(Y<0) THEN ZFPT=PAI(1.5):RETURN
1250 ZFPT=ATN(Y/X)
1260 IF(X>0)AND(Y>=0)THEN RETURN
1270 IF(X>0)AND(Y<0) THEN ZFRT=PAI(2)+ZFPT:RETURN
1280 ZFPT=PAI(1)+ZFPT:RETURN
1290 '
1300 '          直交座標形式への変換
1310 '
1320 LABEL"RECTANGULAR_FORM":ZFR=XRADIUS*COS(THETA)
1330 ZFI=XRADIUS*SIN(THETA):RETURN
1340 '
1350 '          複素数の積(極座標による)
1360 '
1370 LABEL"POLAR_ZMUL":X=XR:Y=XI:GOSUB"POLAR_FORM"
1380 R1=ZFPR:T1=ZFPT:X=YR:Y=YI:GOSUB"POLAR_FORM"
1390 R2=ZFPR:T2=ZFPT:XRADIUS=R1*R2:THETA=T1+T2
1400 GOSUB"RECTANGULAR_FORM":RETURN
1410 '
1420 '          複素数の商(極座標による)
1430 '
1440 LABEL"POLAR_ZDIV":X=XR:Y=XI:GOSUB"POLAR_FORM"
1450 R1=ZFPR:T1=ZFPT:X=YR:Y=YI:GOSUB"POLAR_FORM"
1460 R2=ZFPR:T2=ZFPT:XRADIUS=R1/R2:THETA=T1-T2
1470 GOSUB"RECTANGULAR_FORM":RETURN
1480 '
1490 '          複素数のベキ乗(極座標による)
1500 '
1510 LABEL"ZPOW":X=XR:Y=XI:GOSUB"POLAR_FORM"
1520 R1=ZFPR:T1=ZFPT:XRADIUS=R1^N:THETA=T1*N
1530 GOSUB"RECTANGULAR_FORM":RETURN
1540 '
1550 '          複素数のベキ乗根(極座標による)
1560 '
1570 LABEL"ZNROOT":X=XR:Y=XI:GOSUB"POLAR_FORM"
1580 R1=ZFPR:T1=ZFPT
1590 XRADIUS=R1^(1/N):THETA=(T1+PAI(2)*ITH)/N
1600 GOSUB"RECTANGULAR_FORM":RETURN
1610 '
1620 '          2次方程式の解法
1630 '
1640 LABEL"DIM2_EQUATION":D=B*B-4*C
1650 XR=D:XI=0:N=2:ITH=0:GOSUB"ZNROOT":X1=ZFR/2:Y1=ZFI/2
1660 XR=D:XI=0:N=2:ITH=1:GOSUB"ZNROOT":X2=ZFR/2:Y2=ZFI/2
1670 XR=-B/2:XI=0:YR=X1:YI=Y1:GOSUB"ZADD":RR(0)=ZFR:RI(0)=ZFI
1680 XR=-B/2:XI=0:YR=X2:YI=Y2:GOSUB"ZADD":RR(1)=ZFR:RI(1)=ZFI
1690 RETURN
1700 '
1710 '          3次方程式の解法
1720 '
1730 LABEL"DIM3_EQUATION":P=C-B*B/3:Q=2*B*B*B/27-B*C/3+D
1740 BB=B:CC=C:B=Q:C=-P*P*P/27:GOSUB"DIM2_EQUATION"
1750 U3R=RR(0):U3I=RI(0):V3R=RR(1):V3I=RI(1)
1760 IF(U3R<>0)OR(U3I<>0) THEN GOTO 1810
1770 UR=0:UI=0
1780 IF(V3R<>0)OR(V3I<>0) THEN GOTO 1800
1790 VR=0:VI=0:GOTO 1830
1800 XR=V3R:XI=V3I:N=3:ITH=0:GOSUB"ZNROOT":VR=ZFR:VI=ZFI:GOTO 1830
1810 XR=U3R:XI=U3I:N=3:ITH=0:GOSUB"ZNROOT":UR=ZFR:UI=ZFI
1820 XR=-P/3:XI=0:YR=UR:YI=UI:GOSUB"ZDIV":VR=ZFR:VI=ZFI
1830 W1R=-.5:W1I=SQR(3)/2:W2R=W1R:W2I=-W1I
1840 XR=UR:XI=UI:YR=VR:YI=VI:GOSUB"ZADD"
1850 RR(0)=ZFR-BB/3:RI(0)=ZFI
1860 XR=UR:XI=UI:YR=W1R:YI=W1I:GOSUB"ZMUL":T1R=ZFR:T1I=ZFI
1870 XR=VR:XI=VI:YR=W2R:YI=W2I:GOSUB"ZMUL":T2R=ZFR:T2I=ZFI
1880 XR=T1R:XI=T1I:YR=T2R:YI=T2I:GOSUB"ZADD"
1890 RR(1)=ZFR-BB/3:RI(1)=ZFI
1900 XR=UR:XI=UI:YR=W2R:YI=W2I:GOSUB"ZMUL":T1R=ZFR:T1I=ZFI
1910 XR=VR:XI=VI:YR=W1R:YI=W1I:GOSUB"ZMUL":T2R=ZFR:T2I=ZFI
1920 XR=T1R:XI=T1I:YR=T2R:YI=T2I:GOSUB"ZADD"
1930 RR(2)=ZFR-BB/3:RI(2)=ZFI:RETURN
1940 '
1950 '          検算プログラム（3次方程式）
1960 '
1970 LABEL"DIM3_CHECK"
1980 FOR I=0 TO 2
1990 PRINT "第";I+1;"の解:";RR(I);"+";RI(I);"i"
2000 XR=RR(I):XI=RI(I):N=3:GOSUB"ZPOW":KR=ZFR:KI=ZFI
2010 XR=RR(I):XI=RI(I):YR=XR:YI=XI:GOSUB"ZMUL"
2020 KR=KR+ZFR*B:KI=KI+ZFI*B
2030 KR=KR+RR(I)*C+D:KI=KI+RI(I)*C
2040 COLOR 6
2050 PRINT "方程式の値:";KR;"+";KI;"i"
2060 COLOR 7:NEXT I
2070 RETURN
```

とんでもなくデタラメな話

いつも私たちの気がつかないところで活躍している「乱数」。ここでは少しだけ，この正体不明の数列の真実の姿に触れてみましょう。普段なにげなく使っている乱数はどのように作られているのか。また，正しい乱数はどのように作るべきなのか，といった疑問にお答えします。

Kamon Masato
華門 真人

乱数というものについて考えてみましょう。コンピュータにとって乱数というものは，ゲーム（アクションゲームやRPGはもちろんのことだが，乱数を使ったアドベンチャーゲームというものもあった）やシミュレーションでは欠かすことのできないものです。

特にゲームなどでは高速に乱数を求めるため特殊な方法がとられていることがあります。もっとも簡単なのはZ80のRレジスタを使うものです。Rレジスタの値は0～127までの値を周期的に繰り返しています。ゲームなどでは，

LD A,R

というものを疑似的に乱数と見たてているのです。実際には全然乱数ではないのですが，場合によってはこれで十分，用が足りてしまいます。Z80マシンでは乱数系列の初期化にはこのレジスタを使うことが多く，いつでも新鮮な乱数が発生しますね。こんなレジスタは，ほかのCPUではまずお目にかかれませんから，X68000のX-BASICでユーザーが乱数系列の初期化をしなければならないことにムッとした人もいることでしょう。

このように普段なにげなく使っている乱数ですが，その実体は意外と知られていないようです。

乱数の定義

さて，現実的な問題として乱数(列)とはいったいなんなのでしょうか？　よく使われる定義に「数列の中の数同士になんらの関係も認められないような数列」というものがあります。かみくだいていえば，次になにが出てくるかわからない数のことです。乱数(列)という表現をしましたが，厳密に考えると，ここで考えなければならないのは乱数列なのです。たとえば，5という数は乱数でしょうか。もちろんノーです。乱数という概念は単独で存在するわけではなく，ほかの数とのかかわりで存在するものなのです。ですから当然乱数列という数列単位で考えることが必要になるのです。しかし以下ではこのことを暗黙の了解として，特別な場合を除いて乱数列のことを乱数と表記することにしましょう（そのほうが親しみやすいですからね）。

さて，その乱数ですがもっと親しみやすい表現でいえば，「デタラメ」な数です。身近な例ではサイコロなどがありますね。サイコロを振ると必ず1から6までの目のうちのひとつが出ますが，そのうちのどの目が出てくるのかはまったく予想できません。1回目に2という目が出て，2回目に5という目が出たとしてもこの2と5の間にはなんら関係がないのです。

乱数の発生――その現実

さて，乱数はコンピュータ上にも登場します。BASICのRND関数などがこれにあたりますね。HuBASICなどでは，

RND(1)

というかたちで0以上1未満の乱数を発生することになっています。ですから，コンピュータ上でサイコロを作りたければ，

INT(RND(1)*6+1)

とすればいいわけです。

でも，このRND関数は本当に乱数なのでしょうか？　答えはもちろんノーです。コンピュータにおける乱数は実は疑似乱数と呼ばれるもので，乱数に似せたものではあっても，本当の乱数では決してないのです。

ところでなぜ「もちろん」ノーといったのかわかりますか。乱数の定義のところでも述べたように，乱数とは本質的に互いになんら関係がない存在ですから，乱数を数式で表現することは不可能なのです。一方，コンピュータとは命令（すなわち数式）に従ってのみ動くものであり，数式で表すことのできないものを計算することができるはずがありません。

これについては，かの有名なフォン・ノイマン博士も「四則演算によって乱数を作り出そうと試みる者は，いうまでもなく，神に背こうとしているのである」と語っています（そもそも，この人が初めて算術計算で乱数を作り出そうとした）。

それではRND関数を始めとする（疑似）乱数とはいったいなんなのでしょうか。実はこれらは数式に乗らないはずの乱数を無理やり数式に乗せてしまったものなのです。すなわち，あえて「神に背いて」乱数を数式でシミュレートしたものなのです。もちろん，正確な意味では乱数になりません。だから「疑似」乱数なのです。

しかし実際問題として，

INT(RND(1)*6+1)

のサイコロなどはかなり実用に耐えますね。日常使う分には「疑似」であることはまったく意識しなくてすみます（こうした陰にはできるだけ真の乱数に近いものを作ろうとしたプログラマたちの努力があるのですが）。

それでは，コンピュータ上の疑似関数はいったいどんな数式に従って動いているのでしょうか。

乱数テスト

ひとつ面白い実験をしてみましょう。RND関数などの疑似関数がどれだけ真の乱数に近いかを調べるためのテストです。

HuBASIC上でリスト1を実行してみてください。リスト1はひと目見てわかるように非常に単純なプログラムです。単にRND関数でPSETを行っているだけなのですから。もしこのRND関数が本当の乱数で

あるのならば当然どんどんデタラメな点が打たれていき，最後には画面は真っ白になるはずですね。

図1を見てください。これはリスト1を3時間ほど実行し続けてみたものです。ごらんのように画面は真っ白にはならず，模様のようなものを生じてしまいます。この模様はいったいなんなのでしょうか。実は，これはHuBASICで使われている「線形合同法」という疑似乱数発生法の持つ規則性が表れたものなのです。

線形合同法

線形合同法は現在の乱数発生法のうちもっともメジャーなものです。この方法は乱数列〈X_n〉を，

$$X_{n+1}=(aX_n+c) \bmod m$$

m ：法　　(m>0)

a ：乗数　(0≦a<m)

c ：増分　(0≦c<m)

X_0：出発値 ($0≦X_0<m$)

として定めるものです。このように定められる数列〈X_n〉を線形合同数列（linear congruential sequence）と呼びます。

この線形合同数列はmやaなどの値をうまく設定してやればきわめて乱数に近い性質を持つようにすることもできます。しかし，この線形合同法で生成した疑似乱数列には共通した欠点があるのです。すなわち必ず「環にはまり込み」，周期性を持つようになるのです。これは先ほどテストしたとおりです。もっともこれは線形合同法だけの欠点ではなく，$X_{n+1}=f(X_n)$という数列すべての持つ特徴なのですが。

それでは，この線形合同数列は乱数として失格なのでしょうか。もちろん真の乱数列には絶対なりえません（これは不可能）が，疑似乱数として考えた場合，十分に使いものになるのです（もちろんうまくやれば，ですが）。

より真の乱数に近い線形合同数列とはどんな数列でしょうか。最終的には必ず環にはまり込んでしまうとしても，周期を十分に大きくすることで環にはまり込むことを気にさせなくすることはできそうです。たとえば，100個の乱数からなる乱数列を得たいとします。このときは周期が101であれば（すなわち1番目の数と101番目の数が同じになる）その100個に関していえばほとんど乱数とみなすことができます（環にはまり込んでいないという意味で）。

要するに周期が大きければ大きいほど真の乱数に近づくのです。それではその周期を大きくするためにはどうすればよいのでしょうか。

法 m

まず初めに線形合同法の乱数生成式のうち，法mについて考えてみましょう。この結論はきわめて簡単です。いろいろと制限はありますが，一般にはmは大きいほどよいのです。

というのも乱数列〈X_n〉の周期はmを超えることがないからです。なぜなら線形合同数列X_nは一度でも$X_k=X_r$(ただしk>r)となろうものならそれ以降は必ず，$X_{k+1}=X_{r+1}$，$X_{k+2}=X_{r+2}$，$X_{k+3}=X_{r+3}$……となってしまいます。つまり環にはまり込み，周期なるものが生じてしまうわけです。

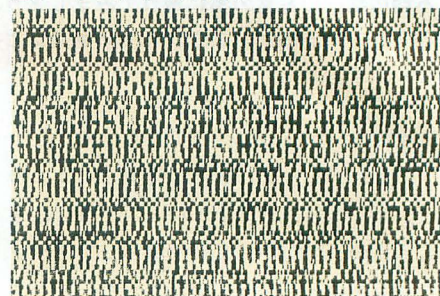
ところがこの周期はmを超えることができないのです。考えてみてください。一般にX_nはmで割った余りですから，mを超えることはないわけです。すなわち，$0≦X_n<m$であり，X_nはm通りの数しかとりえないのです。よってmは大きいほうがよいのです。

とはいえ，mはただ大きければそれでよいというわけではありません。いくら大きくてもあまり乱数にならない場合もありますし（周期は最大mであって，常にmなわけではないですからね），それにヘタに大きくしすぎると，計算が面倒になってしまいます。

それではどんな値が適当なのでしょうか。まずいえるのは，いくら大きいといっても処理できる範囲内で選ぶべきであるということです。オーバーフローしようものなら一気に面倒になりますからね。問題はこの範囲のなかで，どのような値をとるべきかということです。これは後述する乗数aにもかかわることなのですが，結論からいってしまうと，aとmは互いに素（1以外の公約数を持たない）でなければなりません。これは証明しているとかなり長くなって面倒なのですが，簡単にいってしまうとaX_nのとりうる値が少なくなってしまうということです。

リスト1　乱数を調べる

```
10 X=INT(RND(1)*640)
20 Y=INT(RND(1)*200)
30 PSET (X,Y)
40 GOTO 10
```

図1　リスト1の実行例

また，もうひとつ問題があります。それはmを2^eなどの演算語の大きさ（たとえば32ビット演算であれば2^{32}）と同じにすると注意が必要になるということです。$m=2^e$とすると，modの計算が除算なしですむためマシン語での計算が非常に楽になります。反面，得られる乱数の下位ビットがかなり規則的になってしまうという欠点を持ちます。これも説明すると長くなってしまうので，そういうものなのだと思ってください（ずいぶん強引だな～，興味のある人は参考文献を参照のこと）。

だからといって$m=2^e$にしてはいけないというわけではありません。上位ビットは十分にデタラメになっていますから，上位ビットだけを抽出してやれば，乱数として使いものになるのです。要は使い方の問題でしょう。計算の楽さ（これはスピードの速さにもつながる）とある程度の精度を両立させるのもいいし，もし本当に精度を求めるのであれば，mとして2^e-1や2^eにもっとも近い素数を選べばよいのです。実際問題として通常に使う分には$m=2^e$としてやることのほうが多いようです（もちろん上位ビットのみを使いますが）。

乗数 a

今度は最適な乗数aを考えてみましょう。先に述べたように，周期は最大でmです。ですから，最適なaとは周期をmにするaである，ということができます。周期がmになるような，すべてのa，c，X_0の選び方（必要十分条件）は次のようになります。

1)　cとmが互いに素
2)　b=a−1がmを割り切るすべての素数pの倍数である
3)　mが4の倍数であれば，bも4の倍数である

1)はだいたいわかりますね。問題は2)と3)です。実はこれらを満足すれば自動的に「aとmは互いに素である」という条件を満たします。ですから，できるだけ長いmを設定し，それにあわせて1)〜3)を満たすaとcを作ってやればよいわけです。

具体的な例で見てみましょう。mによく使われる値として，2^e, 2^e-1, 2^eにもっとも近い素数の3つがあると先に述べましたが，それぞれについて理想的なaやcの値を考えてみましょう。

m=2^eの場合

1)からcは2^eと互いに素な値，たとえば1，3などの奇数であればよいことになります。また2)および3)からb(=a−1)が4の倍数でなければならないことになります。すなわちaは4で割って1余る(a mod 4=1)数でなければならないことになります。

m=2^e-1の場合

この場合はまずmの約数を探さなければなりません。たとえばm=$2^{36}-1$の場合は約数は3^3, 5, 7, 13, 19, 37, 73, 109ですから，これらの数を約数に持たない数（たとえば4，11など）を選び，それをcとすればよいわけです。また，mはこの型の場合4の倍数にはなりえませんから，3)は考えなくてもすみます。

問題は2)です。2)によりbが3, 5, 7, 13, 19, 37, 73, 109すべての倍数でなければなりません。すなわち，b=n*3*5*7*13*19*37*73*109=7635497415n（nは整数）でなければならないことになります。よって，a=1+7635497415nですね（ただしa<mより0≦n<9）。この場合はうまくaが1以外にも存在しましたが，場合によるとa=1しか存在しないということもあるので注意が必要です。

m=2^eにもっとも近い素数の場合

この場合cはm以外のどんな数でもOKということになります。同様にして3)も考えなくてすみます。問題は2)でこの条件によりa=1と固定されてしまいます。

a=1で周期がmとなり，万事OKとなりそうですが，ところがそうはいかないのです。すなわち，周期はmでもその中でのデタラメさに欠けてしまうのです（$X_{n+1}=(X_n+c) \bmod m$となってしまうのですから）。ここらへんが難しいところですね。

理想的な線形合同法

さて，いろいろと面倒になってきましたので少しまとめてみましょう。$\langle X_n \rangle$がより真の乱数に近くなるためにはできるだけ周期が長いほうがよいのでしたね。

$$X_{n+1}=(aX_n+c) \bmod m$$

では，より長い周期を実現するための条件は次のようになります。

1)　出発値　X_0

X_0は適当でかまいません。もちろん使うたびに発生する乱数が異なるようにするためには毎回X_0を変えなければなりません。そのためには前回使った最終値を覚えておいてそれをX_0としてもよいですし（要するにその数列を永遠に続けることになる），より簡単な方法としてRND関数を使い始めたときの時刻などをX_0にしてしまうという手もあります。

BASICではRANDOMIZEという命令がありますが，この命令はまさしくX_0を自分で設定するための命令です。これをうまく使えば毎回違った乱数列や，あるいは逆に毎回同じ乱数列を生成できるようになるわけです。

2)　法　m

法mはできる限り大きい数を選びます（できれば2^{30}ぐらいはほしい）。m=2^e（計算の語の大きさ）とすると計算が非常に楽になりますが，その代わり先ほども述べたように下位ビットがあまりデタラメでなくなってしまいますので注意が必要でしょう。このような事態を避ける（要するに精度を高くする）ためには，計算は少し面倒になりますが，mとして2^e-1や，2^eにもっとも近い素数を選べばよいわけです。場合に応じて使い分けてください。

3)　乗数　a

周期をmと等しくするためには乗数aのところで述べた3条件を満たさなければなりません。この条件を満たし，なおかつ十分にデタラメな乱数列を生成できるのはm=2^eのときがほとんどです（ほかの場合はaの自由度が少なくa=1となりやすい）。ですから，確実に周期がmで十分にデタラメな乱数列を得たければ，

m=2^e
(cはmと互いに素)
a　mod　4=1

を満たすものを選べばよいことになります。ただし，この場合は上位ビットしか使用してはいけません。

また，m≠2^eの場合は周期がmかつ十分にデタラメとすることが困難（不可能とは限らない）になりますが，その反面下位ビットも十分に使えるようになるというメリットがあります。ですから精度が必要になる場合は少しぐらい周期を小さくしてでも（詳しくは述べませんがうまくやれば周期をm−1とすることすらできます），m≠2^eとすればよいわけです。必要とされる精度によってうまく使い分けてください。

線形合同法の応用

以上で線形合同法の最適な形態を見てきたわけですが，この線形合同法をうまく応用してやればさらにデタラメさを増したり，周期を延ばしてやることが可能になります。この代表的なものが，

$$X_{n+1}=(bX_n^2+aX_n+c) \bmod m$$

なる式です。

このように2次合同法にまで拡張してやればデタラメさも増し，また制約条件もゆるくなります。この数列が最長周期mを持つ条件は次のとおりです。

1)　cとmは互いに素
2)　bとa−1の両方がmを割り切るすべての素数pの倍数
3)　bが偶数であり，mが4の倍数であればbを4で割った余りとa−1を4で割った余りが等しくなければならない。またbが偶数であり，mが2の倍数であればbを2で割った余りとa−1を2で割った余りが等しくなければならない
4)　mが9の倍数であれば，bも9の倍数，もしくはcbを9で割った余りが6となる

また，今度は周期を増す拡張として，X_{n+1}をX_nだけでなく，X_nおよびX_{n-1}の関数とすることが考えられます。こうすれば数列が環にはまり込むのは（X_{n+r}, X_{n+r+1}）=（X_n, X_{n+1}）となるときからになりますから，理論上は最長周期をm^2にすることが

できるようになりますが，この乱数列は実際にはうまくいきません。周期は長くなるものの，デタラメさがなくなってしまうのです。しかし次の式は同じような考え方によって長い周期と十分なデタラメさを実現しています。

$X_n=(X_{n-24}+X_{n-55})\mathrm{mod}\ m$

$(n>=55)$

一見すると，なんなんだこの式はと思われるかもしれませんが，実にうまくできた式で周期が最低でも$2^{55}-1$も保証され，さらにほとんど理論的な裏づけはないのですが，十分にデタラメになっているのです。

このような拡張法によりデタラメさや周期を延ばすことができますが，まだほかにも方法はあります。2つの数列（必ずしも乱数列でなくてもよい）を結びあわせて第3の十分にデタラメで周期の長い乱数列を作るのです。このような乱数生成法をかき混ぜ法と呼びます。

かき混ぜ法1

この方法は2つの数列$\langle X_n\rangle$，$\langle Y_n\rangle$と補助用の表V[0]，V[1]，……，V[k−1]を使います。kは100ぐらいが理想的です。表Vには$\langle X_n\rangle$の最初のk個の値を入れておきます。

手順1　X, Yの生成

X，Yに数列$\langle X_n\rangle$，$\langle Y_n\rangle$の次の項を入れます。

手順2　jの抽出

j=[kY/m]（[r]とはint(r)のこと）とします。このmは$\langle Y_n\rangle$を生成するときに法として使った値ですから，jはYによって決まる$0\leqq j<k$の乱数ということになります。

手順3　交換

V[j]を出力し，V[j]=Xとします。要するに，乱数をまた乱数化するような作業です。この方法の優れたところは乱数列の近い項の間での関係がほとんど完全になくなる点にあります。この方法を使わないと，ある数式に従っている以上，どうしても近い項同士に関連性が生じやすいのです。さらに，この方法では多くの場合に周期を$\langle X_n\rangle$と$\langle Y_n\rangle$の最小公倍数にできるのです。

いいことずくめのようなこの方法ですが，ごくまれに（X_nとY_nにかなり規則性のあるものを使用した場合）この方法によって，より規則性が強い数列を生じてしまうことがあるのです。そこで，次のかき混ぜ法2の登場です。これは前記のかき混ぜ法1を改良したもので，しかも生成法はもっと簡単です。

かき混ぜ法2

これは2数列を使わなくても1数列$\langle X_n\rangle$だけで十分にかき混ぜることができる方法です。この方法でも補助表V[0]，V[1]，……，V[k−1]を使用します。まず初めに表Vのなかに$\langle X_n\rangle$の最初のk項を代入して，補助変数Yにk+1項を代入しておきます（数列Y_nの代わりにYを使うわけです）。

手順1　jの抽出

j=[kY/m]とします。ここでmは$\langle X_n\rangle$を生成するときに法とした数ですから，jはYによって決まる$0\leqq j<k$の乱数ということになるわけです。

手順2　入れ替え

Y=V[j]とし，Yを出力します。V[j]には数列$\langle X_n\rangle$の次の項を代入しておきます。

この方法のほうがはるかに簡単ですね。しかもこの方法は手間がそれほどかからないにもかかわらず，数列をかなりデタラメにしてくれます。もとの数列$\langle X_n\rangle$がかなり規則的であっても出力は十分にデタラメとなるのです。

さて，以上いろいろな乱数の発生法を見てきました。しかしどの方法も基本的には

$X_{n+1}=(aX_n+c)\mathrm{mod}\ m$

という式です。あまりにもいろいろな種類のものを見てきたので少し混乱してしまったかもしれませんね。実際に使う分には$m=2^e$とするもので十分でしょう。この方法ですと，なによりも計算が楽になるというメリットがあります。その代わり下位ビットが規則的になりやすいというデメリットもあるのですが，これは上位ビットを抽出してやればすむことです。

さらにこれにかき混ぜ法2を併用すれば完璧でしょう。実用上は99%までこれで十分です。実用をはるかに超えてとにかく真の乱数を目指すんだという方はほかの方法を使ってください。

乱数列というものは無限の種類を持っています。もちろん発生法も多種多様です。非常に簡単な発生法から，非常に高度な発生法までさまざまです。ですから当然使い分けの必然性が生じてくるのです。モンテカルロ法などの乱数を利用した数値演算にはどんなに複雑になっても高精度な乱数ルーチンが必要になりますし，反対にゲームなどではそれほど精度は必要ないわけです。Z80のRレジスタを使ったものから，かき混ぜ法までいろいろな乱数発生法があります。ぜひ自分でいろいろと試してください。きっと奥深い乱数の世界が見えてくると思います。あなたは「神に背く」ことができるでしょうか。

最後にもっと乱数を究めたいという人のために参考文献を紹介します。この本は『THE ART OF COMPUTER PROGRAMMING』という全14巻にも上る大著の3冊目です。この全集はACMチューリング賞（コンピュータ界のノーベル賞みたいなもの）受賞者であるコンピュータの大家クヌース氏の手になるもので，主に数値演算に関するプログラミングテクニックについて記されています。

コンピュータで数値演算などをやってみたいと思っている人は一度は読んでみることをおすすめします。絶対に役に立ちます。この乱数に関しても今まで述べてきたことだけでなく，乱数の検定など多くのことが述べられています。

上で述べてきたのは乱数に関してのほんのさわりにしかすぎません。ただ乱数列を作っただけではなく検定（本当に正しい乱数列になっているかどうか）もしなければなりませんし，そもそも乱数列がいったいなんなのかというはっきりした定義もしていません。

最初に本当の乱数はコンピュータ上で作ることはできないと書きました。すなわち，「これが正解！」という数式などは存在しないわけです。逆にいえばそれだけいくらでもやりようはあるということになります。たったひとつの乱数を発生させるにも実に多種多様な方法が存在するのです。この決して正解にたどりつくことのない乱数の世界をぜひ自分でも研究してみてください。

参考文献

D.E.クヌース，「準数値算法／乱数」，サイエンス社

特集　真夏の夜の数値演算

そこにπがあるから

π――小学生でも知っている円周率。しかし，このπはあらゆる定数のなかでもっともストロングな数字でもあります。それゆえに，πをめぐってさまざまな人々が知恵をしぼってきました。ここではその苦難の歴史をかいま見てみましょう。

Tan Akihiko
丹　明彦

πに群がる人々

最近，といっても今年の1月末のことだが，某大学のスーパーコンピュータによって，円周率の計算桁数が2億桁を突破したという記事が新聞の片隅に載ったのを覚えている人はいるだろうか。新聞を読まない人は問題外としても，あまりいないような気がする。はっきりいって，実用に向きそうには思えない，ほとんど趣味でやっているような研究に見えるからだろう。実際，通常の科学技術計算では，せいぜい30桁もあれば十分すぎるといわれている。

しかし，(一部の)数学屋さんや(一部の)計算機屋さんにとって，円周率，またはその計算は尽きない興味の的である。なぜなら，πという数は，つっつけばつっつくほどいろんなことがわかってくるし，複雑な長時間の計算は計算機のパワーを計る目安にもなる。そして，円周率にとりつかれた人人が口を揃えていうのだが，「そこにパイがあるから」（と書くとまるで食べ物みたいだな）じゃなくって，「そこにπがあるから」計算するんだ！　ということなのだ。そう，πこそは高精度計算界の最高峰であり，彼らはそこを目指す登山家の心境なのだ。

僕の知人にもそういう人がいる。彼も円周率の魅力にとりつかれたひとりで，ずいぶん昔，そう，MZ-80KやNECの名機PC-8001[1]が最新鋭だった頃から，機会のあるごとに，つまりおよそ計算機と名のつくものに触れるたびに，円周率の計算の腕を磨いていたのである。そして去年，彼は僕のX68000に触ったとき，当然のごとく，このライフワークともいうべき作業に取りかかったのだった。しかし，そのとき使えたのは，BASICインタプリタと，福袋の中のアセンブラ（とリンカ）だけだった。そして彼は，やはり当然のごとくアセンブラに手をつけた……。

それから1カ月もしないうちに，MC68000の命令語やバスエラーやアドレスエラーとの戦い方[2]を覚えた彼は5万桁のπを2時間弱で，1万桁なら4分半で計算するプログラムを作り上げたのだった。スーパーコンピュータのパワーに比べると，こんなのはどうってことない数字だ。2億桁の計算にかかった時間はたったの6時間半なのだから。しかし，これは現時点のパソコンとしてはおそらく最高レベルであろう[3]。僕はCPU (MPU) のパワーを計るのには，下手なベンチマークより，こういう計算をさせたほうがよっぽどいいのではないかと思っている。計算機の計算機としての実力を知るためには，こういうストロングなの[4]がちょうどいい（どこかに，8086[5]でやった人はいないかな)。

πという数は計算機がこの世に出現するはるか昔から数学者たちを虜にしてきた。その性質は謎に満ち，いまだによくわかっていない*。彼らがπにどのようなアプローチを試みたか，駆け足で眺めてみよう。

1) 「名機」と呼んだのには他意があるわけではない。ただ，個人的にNECのマシンで好きなのは唯一あれというだけのことだ。

2) 今から考えると，デバッガもなしでよくやったもんだと思う。僕などは脇から見ていて無責任な助言ばっかりしていたが，内心ひどく感心させられたものだ。

3) ちなみにI/O誌に昔載ったπ計算のプログラムでは，1万桁を計算するのにZ80で1時間50分，6809でも1時間10分かかっている。

4) 要するに，うんざりするほど長く，そして単調な計算。ただし，これを実現しようと思ったら，CPU（MPU）の持っている命令をひととおり使い尽くすくらいの根性がいるらしい。

5) 今はあまり使われていないようだが，68000と開発時期が近い16ビットCPUだから選んだだけのこと。もっとも16ビットで，ということであれば80286までは文句はいえまい。どちらも現在の主力機種に搭載されているのだから。どちらにしても，こういう勝負なら，レジスタ長の大きい68000が有利だろう。アセンブラを使う限りでは，68000はどう見たって32ビットである。だからといって80386を相手に持ってくるのはやっぱり反則だろうと思うが。ベンチマークにしろなんにしろ，CPUやMPUのパワーを比べるときには，フェアプレイなどはありえないのだろう。

アルキメデスの方法

円周率というのは円の直径に対する円周の比で値はご存じ3.1415……である。そこで，その定義に戻って円になるべく近い多角形を作り，その辺の長さの合計を調べるという方法が長い間とられてきた。この方法をとった人のなかでいちばんの有名どころといえば，やはりアルキメデス(Archimedes)である。

＊）πは，無理数で，さらに超越数でもある。

無理数とは無限に続き，しかも循環しない（何桁までいっても繰り返す部分がない）小数のことである。たとえば，$\sqrt{2}=1.41421356\cdots$は無理数である。超越数とは，「代数的でない数」つまり「有理係数の代数方程式の解になりえない数」のこと。そのなんたら方程式というのは，

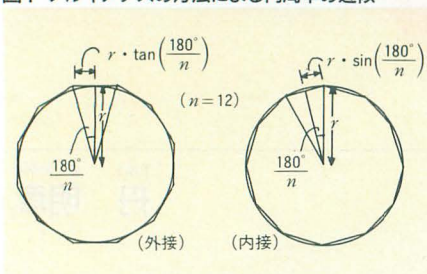
$$a_nX^n+a_{n-1}X^{n-1}+\cdots a_1X+a_0=0$$
$$(a_n,\ a_{n-1},\ \cdots,\ a_0 は有理数)$$

の形で表される方程式。どういうことかといえば，要するに，超越数は加減乗除や平方根を用いるだけでは，値を決められないのである。今回紹介するプログラムでは，そういう簡単な演算しか使ってないが，ある数列が，πに，誤差を無視できる程度に近づくまで計算を繰り返しているだけのことで，別にπ＝……という形で計算式ができていて，その値を計算してるわけではない。

ほかにも，有名な超越数の例として自然対数の底eやオイラー定数のがある。少なくとも理系の人にはお馴染みであろう。eも，近似計算しかできないが，ついでに計算式を掲げておこう。ちょっとした練習問題にはなるだろう。

$$e=\sum_{n=0}^{\infty}(1/n!)$$
$$=1/0!+1/1!+1/2!+1/3!+1/4!+\cdots$$
$$=1+1+1/2+1/6+1/24+\cdots$$
$$=2.71828182\cdots$$

$$\gamma=\lim_{n\to\infty}\left(1+\frac{1}{2}+\frac{1}{3}+\cdots+\frac{1}{n}-\log n\right)$$
$$=0.57721566\cdots$$

図1 アルキメデスの方法による円周率の近似

半径rの円の外接多角形と内接多角形を考える。ちょっと考えればわかるが，辺の長さの総和はそれぞれ，

(外周) $= n \cdot 2\gamma \tan(180°/n)$

(内周) $= n \cdot 2\gamma \sin(180°/n)$

である（図1）。さて，nが大きくなっていくと，この2つの値は$2\pi\gamma$に，それを狭むようにして近づいていく。つまり，

$n \cdot \sin(180°/n) < \pi < n \cdot \tan(180°/n)$

であり，$n \to \infty$（無限大）のとき，

$n \cdot \sin(180°/n) = \pi = n \cdot \tan(180°/n)$

となるはずだ。ということは，もし無限の辺を持つ正多角形の周長が計算できれば，πの正確な値が手に入ることになる。

もちろん三角関数などなかった昔のこと，正接（tan）や正弦（sin）などの値は三平方の定理などを活用する幾何学的手法に頼るしかなかったのだ。アルキメデスは正6角形から出発して辺の数を倍々と増やし，正96角形に達したところで，

$3 + 10/71 < \pi < 3 + 1/7$

とした。小数に直してみよう。

$3.1408\cdots < \pi < 3.1428\cdots$

まだやっと小数点以下第2位までである。

ではだいたい何角形まで取ればまともな値が出るのだろう。リスト1を実行してみてほしい。nの値を聞かれるので適当な数を入れると，外接，内接正n角形を使って円周率を近似してくれる。

リストの中を見るとsin( )やtan( )は平気で使っているし，確かπを近似するプログラムだったはずなのに，pi( )つまりシステムが持っているπの値を角度の計算に使っている。はっきりいって邪道であるが，まああくまでテストであるから気にしない。$n=100000000$（正1億角形！）くらいでだいたいπになったことと思う。

ちなみに，ドイツではπのことを「ルドルフ（Ludolph）の数」と呼ぶが，これはこの方法を使ってπの値を32桁まで計算したオランダの数学者の名前からきている。彼はこれだけの精度を得るのに正2^{62}角形，つまり約10^{18}個の辺を持つ正多角形を作り，計算に一生を費やしたという話だ。

この方式は初等幾何の知識だけで理解できるし，感覚的にもわかりやすいが，高い精度を得るのにはちと無理があるようである。この状態に革新が起きるには17世紀を待たなければならなかったのである。

逆正接関数の公式による計算

この辺は理系の人でないとよくわからないだろうから，心して読むように。

逆正接（アークタンジエント）（arctan）とは，正接（タンジエント）（tan）の逆関数である，なんていっても高校生以下の人にはよくわからないだろうから，もう少しわかりやすくいおう。

角度xの正接をyとする，すなわち，

$y = \tan(x)$

のとき，

$x = \arctan(y)$

の関係が成り立つというのだ，要するに。あまりわかりやすくならなかったが，一応そういうことだ。

で，これが円周率の計算にどう役立つかということを説明しよう。

正接というのは，要するに傾きだ。45°の角度を持つ直線の傾きは1。ちょっと高級な数学では，角度をラジアンで表す。なお，

リスト1 アルキメデスの方法

```
10 /* calculation of pi (1)
20 /*   Archimedes' method
30  float Po, Pi, n, theta
40  input "n=",n
50  theta=pi()/n
60  Po=n*tan(theta): Pi=n*sin(theta)
70  print using "PI(外)=#.############   PI(内)=#.############"; Po, Pi
80 goto 40
```

リスト2 逆正接によるπの計算

```
10 /* calculation of pi (2)
20 /*   formulas by arctangent
30 /*
40 /* a) Gregory & Leibniz
50  print: print using "pi=#.############"; 4#*atan(1#)
60 /* b) Machin
70  print: print using "pi=#.############"; 16#*atan(1#/5#)-4#*atan(1#/239#)
80 /* c) Gauss
90  print: print using "pi=#.############"; 48#*atan(1#/18#)+32#*atan(1#/57#)-20#*atan(1#/239#)
100 /* d) Stormer
110  print: print using "pi=#.############"; 24#*atan(1#/8#)+8#*atan(1#/57#)+4#*atan(1#/239#)
```

リスト3 逆正接による多桁演算

```
10 /* calculation of pi (3)
20 /*   formulas by arctangent (infinite series)
30 /*
40 float p, lim=0.01#
50 /* a) Gregory & Leibniz
60   print space$(15); time$
70   print "Gregory & Leibniz"
80   p=at(1#,4#)
90   print: print using "pi=#.############"; p
100 /* b) Machin
110   print space$(15); time$
120   print "Machin"
130   p=at(1#/5#,16#)-at(1#/239#,4#)
140   print: print using "pi=#.############"; p
150 /* c) Gauss
160   print space$(15); time$
170   print "Gauss"
180   p=at(1#/18#,48#)+at(1#/57#,32#)-at(1#/239#,20#)
190   print: print using "pi=#.############"; p
200 /* d) Stormer
210   print space$(15); time$
220   print "Stormer"
230   p=at(1#/8#,24#)+at(1#/57#,8#)+at(1#/239#,4#)
240   print: print using "pi=#.############"; p
250   print space$(15); time$
260 end
270 func float at( x;float, a;float )
280   float t, x2, y, s=1#, b=1#, arctan=0#
290   int n=0
300   t=x*a: x2=pow(x,2#)
310   repeat
320     y=t/b
330     arctan=arctan+y*s
340     t=t*x2
350     b=b+2#
360     s=-s
370     n=n+1
380   until y<lim
390   print "  n=";n;
400   return(arctan)
410 endfunc
```

(角度) ×π/180＝ (ラジアン) だから, 45°をラジアンに直せばπ/4になる。ということは,

$\tan(\pi/4) = 1$

だから,

$\arctan(1) = \pi/4$

になることは簡単にわかると思う。

それから, 証明をしないで紹介するのは非常に心苦しいのだが, 逆正接関数は次のように級数**の形で表せることも, もっと高等な数学のおかげでわかっている。

$\arctan(x)$
$= x - x^3/3 + x^5/5 - x^7/7 + \cdots$

つまり, πの値は次のようにして求めることができる。

a $\pi = 4\arctan(1)$
$= 4(1-1/3+1/5-1/7+\cdots)$

これを見いだしたのはグレゴリー(Gregory)とライプニッツ(Leibniz)[6]。これを見る限り, 無理数であるπも, しょせんは簡単な分数の和にすぎない。そういうわけで, 計算機の出番となる。それはあとのお楽しみ。もっとも, 水を差すようで悪いが, あとで話すように, この公式は使いものにならないため, もっと使える公式が後世になって現れることになった。

b マチン (Machin) の公式

$\pi = 16\arctan(1/5) - 4\arctan(1/239)$
$= 16(1/5 - 1/3\cdot5^3 + 1/5\cdot5^5 - \cdots)$
$- 4(1/239 - 1/3\cdot239^3 + 1/5\cdot239^5 - \cdots)$

c ガウス (Gauss) の公式

$\pi = 48\arctan(1/18) + 32\arctan(1/57)$
$- 20\arctan(1/239)$
$= 48(1/18 - 1/3\cdot18^3 + 1/5\cdot18^5 - \cdots)$
$+ 32(1/57 - 1/3\cdot57^3 + 1/5\cdot57^5 - \cdots)$
$- 20(1/239 - 1/3\cdot239^3 + 1/5\cdot239^5 - \cdots)$

d シュテルマー (Stφrmer) の公式

$\pi = 24\arctan(1/8) + 8\arctan(1/57)$
$+ 4\arctan(1/239)$
$= 24(1/8 - 1/3\cdot8^3 + 1/5\cdot8^5 - \cdots)$
$+ 8(1/57 - 1/3\cdot57^3 + 1/5\cdot57^5 - \cdots)$
$+ 4(1/239 - 1/3\cdot239^3 + 1/5\cdot239^5 - \cdots)$

このうちbとdについては証明を載せておこう。特に, dの証明の中にはゴチャゴチャしてあふれかえった数字が劇的に消え失せるところがある。よくこんな証明ができたもんだと思うのだが, とにかく, これらの公式が正しいのは, リスト2を実行すればわかるだろう。

さて, この4つの公式が計算をする上でどのように有利か, または不利かを説明してみよう。

a: 形はいちばん簡単なのだが, この級数の収束はとんでもなく遅いのである。簡

図2-1 Machinの公式の証明

b Machinの公式

$\pi = 16\arctan\frac{1}{5} - 4\arctan\frac{1}{239}$

$\tan\frac{\pi}{4} = 1$

$4\times11^\circ = 44^\circ < \frac{\pi}{4} = 45^\circ < 48^\circ = 4\times12$

$\tan\alpha = \frac{1}{5}$★とおくと, $11^\circ < \alpha < 12^\circ$, $\alpha = \arctan\frac{1}{5}$

$4\alpha + \beta = \frac{\pi}{4}$ …(*) とおく,

$\beta = \frac{\pi}{4} - 4\alpha = \frac{\pi}{4} - 4\arctan\frac{1}{5}$

$\tan\beta = \tan\left(\frac{\pi}{4} - 4\arctan\frac{1}{5}\right)$

$= \dfrac{\tan\frac{\pi}{4} - \tan\left(4\arctan\frac{1}{5}\right)^{★★}}{1 + \tan\frac{\pi}{4}\cdot\tan\left(4\arctan\frac{1}{5}\right)^{★★★}}$

$= \dfrac{1 - \left(1 + \frac{1}{119}\right)}{1 + 1\cdot\frac{120}{119}} = -\frac{1}{239}$

$\therefore \beta = -\arctan\frac{1}{239}$

(*)より $\pi = 16\alpha + 4\beta$

$= 16\arctan\frac{1}{5} - 4\arctan\frac{1}{239}$

(終)

(基本的なこと)

$\tan(\arctan x) = x$, $\tan(-x) = -\tan x$

$\tan(x+y) = \dfrac{\tan x + \tan y}{1 - \tan x\cdot\tan y}$ (加法定理)

$\tan 2x = \dfrac{2\tan x}{1 - \tan^2 x}$ (上で $x = y$ とおく)

★

$\tan 11^\circ = 0.194\cdots$
$\tan 12^\circ = 0.212\cdots$から
$\frac{1}{5} = 0.2$を仮定している。

★★

$\tan\left(4\arctan\frac{1}{5}\right)$

$= \dfrac{2\tan\left(2\arctan\frac{1}{5}\right)^{★★★}}{1 - \tan^2\left(2\arctan\frac{1}{5}\right)}$

$= \dfrac{2\cdot\frac{5}{12}}{1 - \left(\frac{5}{12}\right)^2} = \frac{120}{119} = 1 + \frac{1}{119}$

★★★

$\tan\left(2\arctan\frac{1}{5}\right)$

$= \dfrac{2\tan\left(\arctan\frac{1}{5}\right)}{1 - \tan^2\left(\arctan\frac{1}{5}\right)}$

$= \dfrac{2\cdot\frac{1}{5}}{1 - \left(\frac{1}{5}\right)^2} = \frac{5}{12}$

図2-2 Stφrmerの公式の証明

d Stφrmerの公式

$\pi = 24\arctan\frac{1}{8} + 8\arctan\frac{1}{57} + 4\arctan\frac{1}{239}$

$\tan\frac{\pi}{4} = 1$

$6\times7^\circ = 42^\circ < \frac{\pi}{4} = 45^\circ < 48^\circ = 6\times8^\circ$

$\tan\alpha = \frac{1}{8}$★とおくと, $7^\circ < \alpha < 8^\circ$, $\alpha = \arctan\frac{1}{8}$

$6\alpha + \theta = \frac{\pi}{4}$ …(*) とおく,

$\theta = \frac{\pi}{4} - 6\alpha = \frac{\pi}{4} - 6\arctan\frac{1}{8}$

$\tan\theta = \dfrac{\tan\frac{\pi}{4} - \tan\left(6\tan^{-1}\frac{1}{8}\right)^{★★}}{1 + \tan\frac{\pi}{4}\cdot\tan\left(6\tan^{-1}\frac{1}{8}\right)}$

$= \dfrac{1 - \frac{186416}{201663}}{1 + 1\cdot\frac{186416}{201663}} = \frac{15247}{388079}$

$\theta = \arctan\frac{15247}{388079} = 2.249\cdots^\circ$

$\tan\beta = \frac{1}{57}$★★★とおくと, $\beta = \arctan\frac{1}{57} = 1.005\cdots^\circ$

$\theta = 2\beta + \gamma$ とおく,

$\gamma = \theta - 2\beta$

$\tan\gamma = \dfrac{\tan\theta - \tan 2\beta^{★★★★}}{1 + \tan\theta\cdot\tan 2\beta}$

$= \dfrac{\frac{15247}{388079} - \frac{114}{3248}}{1 + \frac{15247}{388079}\cdot\frac{114}{3248}} = \frac{1}{239}$ ← 注目

$\therefore \gamma = \arctan\frac{1}{239}$

(*)より, $\pi = 24\alpha + 4\theta = 24\alpha + 4(2\beta + \gamma)$

$= 24\arctan\frac{1}{8} + 8\arctan\frac{1}{57} + 4\arctan\frac{1}{239}$

(終)

★★★

θの値から見当をつけているのだろう。試行錯誤の末に決めた数と考えられる。

★

$\tan 7^\circ = 0.122\cdots$
$\tan 8^\circ = 0.140\cdots$から
$\frac{1}{8} = 0.125$を仮定

★★

$\tan\left(2\arctan\frac{1}{8}\right)$

$= \dfrac{2\tan\left(\arctan\frac{1}{8}\right)}{1 - \tan^2\left(\arctan\frac{1}{8}\right)} = \dfrac{2\cdot\frac{1}{8}}{1 - \left(\frac{1}{8}\right)^2} = \frac{16}{63}$

$\tan\left(4\arctan\frac{1}{8}\right)$

$= \dfrac{2\tan\left(2\arctan\frac{1}{8}\right)}{1 - \tan^2\left(2\arctan\frac{1}{8}\right)}$

$= \dfrac{2\cdot\frac{16}{63}}{1 - \left(\frac{16}{63}\right)^2} = \frac{2016}{3713}$

$\tan\left(6\arctan\frac{1}{8}\right)$

$= \dfrac{\tan\left(2\arctan\frac{1}{8}\right) + \tan\left(4\arctan\frac{1}{8}\right)}{1 - \tan\left(2\arctan\frac{1}{8}\right)\cdot\tan\left(4\arctan\frac{1}{8}\right)}$

$= \dfrac{\frac{16}{63} + \frac{2016}{3713}}{1 - \frac{16}{63}\cdot\frac{2016}{3713}} = \frac{186416}{201663}$

★★★★

$\tan 2\beta$

$= \tan\left(2\arctan\frac{1}{57}\right)$

$= \dfrac{2\tan\left(\arctan\frac{1}{57}\right)}{1 - \tan^2\left(\arctan\frac{1}{57}\right)}$

$= \dfrac{2\cdot\frac{1}{57}}{1 - \left(\frac{1}{57}\right)^2} = \frac{114}{3248}$

単にいうと，1桁余分に求めようと思えば10倍の時間が必要なのである。このやり方では，1万桁求めようとしても，とてもじゃないが無理で，もしかするとこの世の終わりまでかかってしまうかもしれない[7]。

逆正接関数の値を級数展開で求めるにはarctan( )の中身が小さければ小さいほど収束**が速い。この値は各項に3乗，5乗，……と掛かっていくので，項の値が必要とされている精度より小さくなるのがより速くなるからだ。この観点からも，残りの公式を見てみよう。

[b]：1/5というのは，かなり大きい値なので，この級数も収束が遅めだ（[a]の級数ほどではない）。その代わり，計算式の要素arctan( )は2つだけなので，その分少なくてすむ。速くなることが期待できそうだ。

[c]：要素は3つあるがどの項もarctan( )の中身が小さく，収束もそこそこ速いだろう。

[d]：これも要素が3つ。しかもarctan(1/8)の項は収束が遅そうに見える。確かにふつうにやれば収束は遅い。しかし，コンピュータにとって，$1/8^2$を掛ける（これは，級数の項が小さくなっていくペース），つまり64で割るということにはどういう意味があるだろう。そう，6ビット分シフトすることと同じなのである。これはマシン語をやるとよくわかるのだが，コンピュータは割り算よりはビット演算のほうが得意で，しかも速い。ここで大きく時間を稼げる。

リスト3を走らせよう。それぞれの公式について，取った項数と，πの近似値が表示される。変数limは精度の目安である。あまり小さくすると（精度をよくすると），いつまでも計算が終わらなくなるので注意。これは公式[a]の収束があまりに遅いせいである。少しlimをリスト中の値より小さく取ってみるとわかるが，[a]の計算に手間取っている。ヘタをすると何時間もかかる。それにつきあいたくない人は1桁小さくしておこう。

とにかく遅い。でもその割には，[b]，[c]，[d]はあっという間に終わってしまう。たとえ，倍精度でlimの値を最小限に取っても一瞬のうちに終わる。はっきりいって，この差は大きい。桁が大きくなればなおさらだ。これが先ほど公式[a]が使いものにならないといった理由である。おわかりいただけただろうか。

さて，ここでギンギンに最適化をかけ，高速化を図った，例の去年作られたマシン語のプログラム（リスト4）をお目にかけよう。このプログラムには多くの大技，小技，アクロバット的な技も含めたハイテクニックがつまっている。このリストは公式[d]（シュテルマーの公式）用だが，少し変えれば[b]，[c]もできる。計算桁数も変えられる。その変更法はいっても面倒なだけだからいわない。ソースリスト中のラベルがすべてを語ってくれるだろう（か？）。

とにかく，このプログラムは約2時間かかって5万桁のπの値を計算するのだ。参考までに，[b]，[c]について，1万桁計算させたときの所要時間を表1に挙げておく。

結局，このプログラムでは[d]がもっとも速くできるという結論が得られた（だから採用されているのだが）。といっても，これはarctan(1/8)をビットシフトで計算した分で稼いでいるのであって，それがなければどれも同程度の速さである。この部分は速いだけに相当に技巧的で，マシン語を相当知っている人でも解読するのは困難かもしれない。

なお，アルゴリズムの都合上，計算桁数には限界がある。それを表2に示す。収束が遅いと，項をたくさん取らねばならないので，そこに制限が出るのである。arctan( )の級数の分母の係数は，1，3，5，……となっているだろう。これがどんどん大きくなっていって，16ビットを超えると計算できないのだ。理由を説明しよう。これはマシン語レベルの問題だ。68000のマシン語には，割り算もちゃんとある。しかし，32ビットのデータを16ビットのデータで割るのが限界だから，収束が遅い＝項がたくさん出る＝収束する前に係数が16ビットを超えるような級数はこのプログラムでは計算できないことになるのだ。たとえば，8万桁くらい出そうと思ったら，ガウスの公式[c]を使って6時間半ほどかければいいのだが，シュテルマーの公式[d]では，どんなにがんばっても8万桁なんて出せない。さらにマチンの公式[b]では，5万桁だって出せない。どんなに長い桁数でも計算できる一般的なプログラムは，あればあるに越したことはないが，メモリ容量や計算時間などを考えると，むしろこっちの選択が正しかったと思う。

まだまだ難しい話は続くが，この手の話は，スピードと効率と正確さをギリギリまで追求する円周率の計算では避けて通れないからしかたがない。逆正接関数のアルゴリズムでは，計算時間はおおよそ計算桁数

表1　X68000によるπ計算の所要時間

[b]	Machin	$\pi=16\arctan\frac{1}{5}-4\arctan\frac{1}{239}$	6分17秒
[c]	Gauss	$\pi=48\arctan\frac{1}{18}+32\arctan\frac{1}{57}-20\arctan\frac{1}{239}$	6分06秒
[d]	Størmer	$\pi=24\arctan\frac{1}{8}+8\arctan\frac{1}{57}+4\arctan\frac{1}{239}$	4分38秒（6分23秒）※

※$\arctan\frac{1}{8}$の計算に，ビットシフトを使って最適化した専用ルーチンを使わず，ふつうのarctanを求めるサブルーチンを使った場合。
計算時間は1万桁のπを求めるのに要した時間と，内部表現（2進小数）を10進小数に変換する時間の和である。どちらも桁数の2乗に比例して時間がかかる。

表2　計算桁数の限界

[b]	Machin	45000
[c]	Gauss	82000
[d]	Størmer	59000

**) 数列について，少し説明しておこう。数列とは，いうまでもなく，

$a_1,\ a_2,\ a_3,\ a_4,\ a_5,\ a_6,\ \cdots,\ a_{n-1},\ a_n,\ \cdots$

のことだ。ちょうどBASICの配列のようなものだ。これをひとからげにして

$\{a_n\}$

と書くこともある。数列の各要素（a_1，a_2，…）を項という。詳しい話は学校の教科書に譲る。数列を理解するのに必要な概念はまだある。まず漸化式（ぜんかしきと読む）とは，前の項を使って次の項を作る規則のようなものだ。たとえば，フィボナッチ数列，

1，1，2，3，5，8，…

の漸化式は，

$a_0=1,\ a_1=1$

$a_n=a_{n-1}+a_{n-2}\ (n\geqq 2)$

である。反復法アルゴリズムなどは，この漸化式である。

数列の和を級数という。項を限りなく足しあわせていき，その和がある一定の値に近づいていったとする。このとき，この級数は収束するという。級数が収束するときには，項の値が次第に0に近づいていく場合が多い。本文中でarctan( )の値は，( )の中身が小さいほどよいといったのは，こういうことである。

リスト4 最適化された50000桁プログラム

```
  1: **************  CALCULATION   OF  PI  **************
  2:
  3:
  4: _EXIT           EQU       $FF00
  5: _GETCHAR        EQU       $FF01
  6: _PUTCHAR        EQU       $FF02
  7: _PRNOUT         EQU       $FF05
  8: _PRINT          EQU       $FF09
  9: *NUMB       EXAMPLE        4156   ;  836 ;  420 ; 172
 ; 36
 10: *NUM10      EXAMPLE       10000   ; 2000 ; 1000 ; 400
 ; 80
 11: NUMB            EQU       20768
 12: NUM10           EQU       50000
 13:
 14: NUMW            EQU       NUMB/2
 15: NUML            EQU       NUMB/4
 16:
 17: N_SPACE         EQU       10
 18: N_BLOCK         EQU       10
 19:
 20:
 21: **************  MAIN  PROGRAM  **************
 22:
 23: MAIN
 24:                 BSR       ATAN_8
 25:                 BSR       ATAN_57
 26:                 BSR       ATAN_239
 27:
 28:                 BSR       BCD_CONVERT
 29:                 BSR       BEEP
 30:                 BSR       PRINT_OUT
 31:
 32:                 DC.W      _EXIT
 33:
 34:
 35: **************  24 ATAN(1/8)  **************
 36:
 37: ATAN_8
 38:                 LEA       WORK2+2,A6
 39:                 MOVE.W    #NUMW,D6
 40:                 MOVEQ     #3,D3
 41:                 MOVEQ     #%00100101,D1
 42:                 MOVE.B    D3,(PI+3)
 43:                 MOVE.W    D3,D5
 44:
 45: A_8_SERIES      BSR       A_8_SER
 46:                 BSR       SUB_PI
 47:                 BSR       A_8_SER
 48:                 BSR       ADD_PI
 49:                 BRA       A_8_SERIES
 50:
 51: A_8_SER
 52: A_8_SET         ROR.W     #6,D3
 53:                 ROR.B     #1,D1
 54:                 BCC       DIV_8
 55:                 CLR.W     (A6)
 56:                 ADDQ.L    #2,A6
 57:                 SUBQ.W    #1,D6
 58:                 BNE       DIV_8
 59:                 ADDQ.L    #4,SP
 60:                 RTS
 61:
 62: **************
 63: DIV_8           MOVEA.L   A6,A2
 64:                 MOVE.L    D3,D0
 65:                 MOVE.W    D6,D2
 66:                 MOVE.W    D6,D7
 67:                 LSR.W     D7
 68:
 69: DIV_8N          DIVU      D5,D0
 70:                 MOVE.W    D0,(A2)+
 71:                 CLR.W     D0
 72:                 DBRA      D2,DIV_8N
 73:
 74:                 ADDQ.W    #2,D5
 75:                 RTS
 76:
 77:
 78: **************  8 ATAN (1/57)  **************
 79:
 80: ATAN_57
 81:                 MOVEQ     #3,D5
 82:                 MOVEQ     #57,D0
 83:                 MOVE.L    #$80000,D1
 84:                 MOVE.L    #57*57,D3
 85:                 BSR       ATAN_23
 86:                 RTS
 87:
 88:
 89: **************  4 ATAN (1/239)  **************
 90:
 91: ATAN_239
 92:                 MOVEQ     #3,D5
 93:                 MOVE.L    #239,D0
 94:                 MOVE.L    #$40000,D1
 95:                 MOVE.L    #239*239,D3
 96:                 BSR       ATAN_23
 97:                 RTS
 98:
 99:
100: **************  2nd & 3rd SERIES  **************
101:
102: ATAN_23
103:                 LEA       WORK1+4,A4
104:                 LEA       WORK2+4,A5
105:                 MOVE.W    #NUMW-1,D6
106:                 MOVE.W    #NUML-1,D7
107:                 MOVEA.L   A4,A1
108:                 MOVEA.L   A5,A2
109:                 MOVE.W    D6,D2
110:
111: SERIES_SET      DIVU      D0,D1
112:                 MOVE.W    D1,(A1)+
113:                 MOVE.W    D1,(A2)+
114:                 CLR.W     D1
115:                 DBRA      D2,SERIES_SET
116:                 BSR       ADD_PI
117:
118: SERIES_23       BSR       SER_23
119:                 BSR       SUB_PI
120:                 BSR       SER_23
121:                 BSR       ADD_PI
122:                 BRA       SERIES_23
123:
124: SER_23          MOVEA.L   A4,A1
125:                 MOVEA.L   A5,A2
126:                 CLR.L     D1
127:                 MOVE.L    D1,D4
128:                 MOVE.W    (A1),D1
129:                 DIVU      D3,D1
130:                 MOVE.W    D1,(A1)+
131:                 BNE       DIV_23
132:                 CLR.W     (A5)
133:                 ADDA.L    #2,A4
134:                 ADDA.L    #2,A5
135:                 MOVEA.L   A5,A2
136:                 SUBQ      #1,D6
137:                 BNE       DIV_23_2
138:                 ADDQ.L    #4,SP
139:                 RTS
140:
141: **************
142: DIV_23          MOVE.W    D1,D4
143:                 DIVU      D5,D4
144:                 MOVE.W    D4,(A2)+
145: DIV_23_2        MOVE.W    D6,D2
146: DIV_23_CONT     MOVE.W    (A1),D1
147:                 DIVU      D3,D1
148:                 MOVE.W    D1,(A1)+
149:                 MOVE.W    D1,D4
150:                 DIVU      D5,D4
151:                 MOVE.W    D4,(A2)+
152:                 DBRA      D2,DIV_23_CONT
153:
154:                 MOVE.W    D6,D7
155:                 LSR.W     D7
156:                 ADDQ.W    #2,D5
157:                 RTS
158:
159:
160: ****************************************
161:
162: ADD_PI          LEA       WORK2+NUMB+4,A2
163:                 LEA       PI+NUMB+4,A0
164:                 MOVE.W    D7,D4
165:                 SUB.B     #0,D4
166: ADD_LOOP        ADDX.L    -(A2),-(A0)
167:                 DBRA      D4,ADD_LOOP
168:                 BCC       ADD_END
169: ADD_OV          ADDQ.B    #1,-(A0)
170:                 BCS       ADD_OV
171: ADD_END         RTS
172:
173: **************
174: SUB_PI          LEA       WORK2+NUMB+4,A2
175:                 LEA       PI+NUMB+4,A0
176:                 MOVE.L    D7,D4
177:                 SUB.B     #0,D4
178: SUB_LOOP        SUBX.L    -(A2),-(A0)
179:                 DBRA      D4,SUB_LOOP
180:                 BCC       SUB_END
181: SUB_BORR        SUBQ.B    #1,-(A0)
182:                 BCS       SUB_BORR
183: SUB_END         RTS
184:
185:
186: **************  CONVERT TO DECIMAL CODE  **********
187:
188: BCD_CONVERT
189:                 MOVE.W    #NUM10/20,D7
190:                 LEA       PI+2,A2
191:                 LEA       PI_PRT,A3
192:                 MOVE.B    1(A2),(A3)
193:                 ADDI.B    #$30,(A3)+
194:                 CLR.W     (A2)
195:                 MOVE.W    #10000,D1
196:                 LEA       PI+NUMB+4,A6
197:                 MOVE.W    #NUMW,D6
198:
199: B_C_LOOP        MOVEQ.L   #4,D4
200: B_C_WORD        MOVEA.L   A6,A5
201:                 MOVE.W    D6,D5
202:                 BSR       MULT_CONV
203:                 DBRA      D4,B_C_WORD
204:                 SUBQ.L    #8,A6
205:                 SUBQ.W    #4,D6
206:                 DBRA      D7,B_C_LOOP
207:
208:                 RTS
209:
210: **************
211: MULT_CONV       CLR.L     D2
212: MULT_CONT       MOVE.W    -(A5),D0
213:                 MULU.W    D1,D0
214:                 ADD.L     D2,D0
215:                 MOVE.W    D0,(A5)
216:                 SWAP      D0
217:                 MOVE.W    D0,D2
218:                 DBRA      D5,MULT_CONT
219:
220: CONVERT         MOVEQ.L   #2,D3
221:                 LEA       EXP_TBL+2,A4
222:                 MOVE.W    (A2),D0
223:                 CLR.W     (A2)
224: EX_LOOP0        MOVE.W    (A4)+,D5
225:                 MOVE.B    #'0',D2
226: EX_LOOP1        SUB.W     D5,D0
227:                 BCS       XBCD
228:                 ADDQ.B    #1,D2
229:                 BRA       EX_LOOP1
230: XBCD            ADD.W     D5,D0
231:                 MOVE.B    D2,(A3)+
232:                 DBRA      D3,EX_LOOP0
233:                 ADDI.B    #'0',D0
234:                 MOVE.B    D0,(A3)+
235:                 RTS
236:
237:
238: **************  PRINT  SUBROUTINE  **************
239:
240: PRINT_OUT
241:                 LEA       PI_PRT,A1
242:                 MOVE.B    (A1)+,(INT)
243:
244: INPUT_ORDER     PEA       ORDER
245:                 DC.W      _PRINT
246:                 ADDQ.L    #4,SP
247:                 DC.W      _GETCHAR
248:                 CMPI.B    #'E',D0
249:                 BEQ       GET_E
250:                 CMPI.B    #'e',D0
251:                 BNE       CMP_C
252: GET_E           RTS
253: CMP_C           CMPI.B    #'C',D0
254:                 BEQ       GET_C
255:                 CMPI.B    #'c',D0
256:                 BNE       CMP_P
257: GET_C           MOVE.W    #_PUTCHAR,(PRN_DEST)
258:                 MOVEQ.L   #4,D5
259:                 LEA       TITLEC,A0
260:                 BRA       OUTPUT
261: CMP_P           CMPI.B    #'P',D0
262:                 BEQ       PRINTER
263:                 CMPI.B    #'p',D0
264:                 BNE       INPUT_ORDER
265: PRINTER         MOVE.W    #_PRNOUT,(PRN_DEST)
266:                 MOVEQ.L   #9,D5
267:                 LEA       TITLEP,A0
268:
269: OUTPUT          BSR       PRINT_LINE
270:                 LEA       INT,A0
271:                 BSR       PRINT_LINE
272:                 MOVE.W    #NUM10-1,D7
273:                 CLR.W     D6
274:
275: PUT_PI_1        BSR       FIG_OUT
276:                 MOVEQ.L   #9,D4
277: PUT_PI_2        MOVE.W    D5,D3
278: PUT_PI_3        MOVEQ.L   #9,D2
279: PUT_PI_4        MOVE.B    (A1)+,D0
280:                 BSR       PUT_D0
281:                 DBRA      D7,PUT_CONT
282:                 LEA       BLOCK,A0
283:                 BSR       PRINT_LINE
284:                 BRA       PRINT_OUT
285: PUT_CONT        DBRA      D2,PUT_PI_4
286:                 LEA       SPACE,A0
287:                 BSR       PRINT_LINE
288:                 DBRA      D3,PUT_PI_3
289:                 LEA       LINE_FEED,A0
290:                 BSR       PRINT_LINE
291:                 DBRA      D4,PUT_PI_2
292:                 LEA       BLOCK,A0
293:                 BSR       PRINT_LINE
294:                 BRA       PUT_PI_1
295: PUT_END         LEA       BLOCK,A0
296:                 BSR       PRINT_LINE
297:                 BRA       INPUT_ORDER
298:
299: **************
300: FIG_OUT
301:                 MOVE.W    #' ',D0
302:                 CMPI.B    #4,D5
303:                 BEQ       PRT_C
304:                 MOVEQ.L   #118,D1
305:                 BRA       TAB_PRT
306: PRT_C           MOVEQ.L   #64,D1
307: TAB_PRT         BSR       PUT_D0
308:                 DBRA      D1,TAB_PRT
309:                 LEA       FIG_TAB,A0
310:                 MOVE.W    D6,D1
311:                 ADDQ.W    #1,D1
312:                 BSR       FIG_CONVERT
313:                 SUBQ.W    #1,D1
314:                 ADDA.L    #4,A0
315:                 MOVEQ.L   #100,D2
316:                 MOVE.W    D5,D3
317: FIG_CALC        ADD.W     D2,D6
318:                 DBRA      D3,FIG_CALC
319:                 CMPI.W    #NUM10,D6
320:                 BCS       FIG_NORMAL
321:                 MOVE.W    #NUM10,D6
322: FIG_NORMAL      MOVE.W    D6,D1
323:                 BSR       FIG_CONVERT
324:                 LEA       FIG_TAB,A0
325:                 BSR       PRINT_LINE
326:                 RTS
327:
328: **************
329: FIG_CONVERT     MOVEQ.L   #3,D3
330:                 LEA       EXP_TBL,A2
331:                 MOVE.B    #$20,D4
332: EX_LOP0         CLR.B     D2
333:                 MOVE.W    (A2)+,D0
334: EX_LOP1         SUB.W     D0,D1
335:                 BCS       X_ASC
336:                 ADDQ.B    #1,D2
337:                 BRA       EX_LOP1
338: X_ASC           ADD.W     D0,D1
339:                 TST.B     D2
340:                 BEQ       X_ASC_2
341:                 MOVE.B    #$30,D4
342: X_ASC_2         ADD.B     D4,D2
343:                 MOVE.B    D2,(A0)+
344:                 DBRA      D3,EX_LOP0
345:                 ADD.B     #'0',D1
346:                 MOVE.B    D1,(A0)+
347:                 RTS
348:
349: **************
350: PRINT_LINE      CLR.W     D0
351: P_L_CONT        MOVE.B    (A0)+,D0
352:                 BSR       PUT_D0
353:                 OR.B      D0,D0
354:                 BNE       P_L_CONT
355:                 RTS
356:
357: **************
358: BEEP            MOVE.L    D0,-(SP)
359:                 MOVE.W    #7,-(SP)
360:                 DC.W      _PUTCHAR
361:                 ADDQ.L    #2,SP
362:                 MOVE.L    (SP)+,D0
363:                 RTS
364: **************
365: PUT_D0          MOVE.W    D0,-(SP)
366: PRN_DEST        DS.W      1
367:                 MOVE.W    (SP)+,D0
368:                 RTS
369:
370:
371: ****************************************
372:
373: TITLEC          DC.B      13,10,10,'
 PI  =  ',0
374: TITLEP          DC.B      13,10,10,'
375:                 DC.B      '
 PI  =  ',0
376: INT             DC.B      '  . ',13,10,0,0
377: ORDER           DC.B      13,10,10,' OUTPUT  to
CRT,  PRINTER  or  END  ( C / P / E ) ? ',0,0
378: FIG_TAB         DC.B      '      ~        ',13,10,0
379: SPACE           DC.B      '  ',0,0
380: LINE_FEED       DC.B      13,10,0,0
381: BLOCK           DC.B      13,10,10,0
382:
383: EXP_TBL         DC.W      10000
384:                 DC.W      1000
385:                 DC.W      100
386:                 DC.W      10
387:
388: DUMMY           DS.B      8
389: PI              DS.B      NUMB+8
390: PI_PRT          DS.B      NUM10+8
391: WORK1           EQU       PI_PRT
392: WORK2           EQU       WORK1+NUMB+8
393:
394:
395:                 END
```

の2乗に比例する。たとえば計算桁数を2倍にすれば時間は4倍かかるのである。なぜか。計算桁数すなわち精度を2倍にしようと思えば各arctan( )の級数は項数を2倍取らねばならない。このアルゴリズムでは1項取るごとに一定のペースで（たとえば2桁ずつという風に）精度が上がるからだ。さらに，計算桁数を2倍にすれば足し算などの計算にも2倍の時間がかかる。トータルでは4倍時間がかかることになる。

ここでひとつの限界が見えてきた。どんなにコンピュータの性能が上がっても，桁数をちょっと多くすれば，計算時間はうなぎのぼり。事実，大型機のπ計算競争でも，80年代初めに200万桁のπを100時間以上も計算したあたりで袋小路に入ってしまった。

6) 正統的なドイツ語の読み方では「ライプニッツ」でいいのだが，最近のドイツではどうも「ライブニッツ」という読み方が一般的であるらしい。ということを真面目に論じた文章をこの前読んだ。
7) だいたい10^{10000}（もはや数えようもないが）項の和を取らねばならない。こういうのを天文学的数字というのだろう。

反復法アルゴリズム

まずはなにもいわずにリスト5を実行してみよう。よくわからないが，いつの間にかπができている。

実はこれは，この5年ほどの間，円周率の計算の記録を塗り変えている，まったく新しい計算方式なのである。はっきりいって，僕らのような凡人には，この新しい方式の理論はまったくといっていいほどわからない。これを理解するためには，いや，この理論はまったくわからないものなんだということを悟るためには，ひとりの天才[8]の話をする必要があるだろう。

彼の名はラマヌジャン（Ramanujan）という。なんとなく耳慣れない名だが，彼はインド人の数学者である。多くの天才がそうであるように，小さい頃からとび抜けた能力を発揮しだし，独力で数人の人間が一生かかってもなしえないような仕事をし，そして不幸にも若くして亡くなったのであった。そう，ラマヌジャンは32歳の若さでこの世を去った。ふつうの人間は，32歳までにどれほどの仕事ができるだろうか。それを考えると，彼の偉大さがわかろうというものだ。さらに，これもかなり驚かされることだが，彼は正規の教育をほとんど受けていないのだ。

彼が短い生涯を捧げて研究した内容からは多くの数学的発見が生まれたが，そのなかのひとつとして，最近になって計算機のために適したπの反復計算法アルゴリズムが導き出された。その理屈はとても僕ごときに理解できるようなやさしいものではない[9]ので，詳細を述べることはできないが，とにかく表3に挙げたようなごく簡単な計算の繰り返しでπを得ることができるのである。ほかにもいっぱい公式はあるが，手頃なのを3つ選んでみた。

繰り返すごとに精度が上がっていくのだが，その上がり方がまた凄い。先ほどのarctan( )の級数がどんなに速く収束しても，これにはてんで歯が立たない。残念ながら，そのうまみはリスト5では味わえない。1万桁，10万桁という，ふつうでは考えられ

表3 反復法アルゴリズムによるπの算出

(1) $a_0=1,\ b_0=\frac{1}{\sqrt{2}},\ t_0=\frac{1}{4},\ x_0=1$

十分な精度になるまで

$$\begin{cases} a_{n+1}=\frac{a_n+b_n}{2},\quad b_{n+1}=\sqrt{a_n\cdot b_n} \\ t_{n+1}=t_n-x_n(a_n-a_{n+1})^2,\quad x_{n+1}=2x_n \end{cases}$$

を繰り返す。

$$P_n=\frac{(a_n+b_n)^2}{4t_n}$$

P_nとπが一致する桁数	
$n=0$	0桁
1	2
2	4
3	10
⋮	⋮

(2) $y_0=\sqrt{2}-1,\ \alpha_0=6-4\sqrt{2}$

$$\begin{cases} y_{n+1}=\frac{1-\sqrt{1-y_n^2}}{1+\sqrt{1-y_n^2}} \\ \alpha_{n+1}=(1+y_{n+1})^2\cdot\alpha_n-2^{n+1}\cdot y_{n+1} \end{cases}$$

$$P_n=\frac{1}{\alpha_n}$$

$n=0$	0桁
1	0
2	3
3	8
4	19
⋮	⋮

(3) $y_0=\sqrt{2}-1,\ \alpha_0=6-4\sqrt{2}$

$$\begin{cases} y_{n+1}=\frac{1-\sqrt[4]{1-y_n^4}}{1+\sqrt[4]{1-y_n^4}} \\ \alpha_{n+1}=(1+y_{n+1})^4\cdot\alpha_n-2^{2n+3}\cdot y_{n+1}(1+y_{n+1}+y_{n+1}^2) \end{cases}$$

$$P_n=\frac{1}{\alpha_n}$$

$n=0$	0桁
1	8
2	41
3	171
4	694
⋮	⋮

リスト5 反復法によるπの計算

```
 10 /* calculation of pi (4)
 20 /*   repetition algorithm
 30 /*
 40   float lim=1E-013#
 50   pi_1()
 60   print: /*print
 70   pi_2()
 80   print: /*print
 90   pi_3()
100 end
110 /* part 1
120 func pi_1()
130   float An, An1, Bn, Bn1, Tn, Tn1, Xn, Xn1, P, differ
140   int n=0
150   An=1: Bn=1#/sqr(2#): Tn=1/4#: Xn=1#: P=pow(An+Bn,2#)/(4#*Tn)
160   repeat
170   /*print using "  A#=#.#############  B#=#.#############"; n, An, n, Bn;
180   /*print using "  T#=#.#############  X#=##"; n, Tn, n, Xn
190     print using "p#=#.#############"; n, P: print
200     An1=(An+Bn)/2#: Bn1=sqr(An*Bn)
210     Tn1=Tn-Xn*pow(An-An1,2#): Xn1=2#*Xn
220     P=pow(An1,2#)/Tn1
230     differ=An-Bn: An=An1: Bn=Bn1: Tn=Tn1: Xn=Xn1: n=n+1
240   until differ<lim
250   return()
260 endfunc
270 /* part 2
280 func pi_2()
290   float Yn, Yn1, An, An1, P, differ
300   int n=0
310   Yn=1#/sqr(2#): An=1#/2#: P=1#/An
320   repeat
330   /*print using "  Y#=#.#############  A#=#.#############"; n, Yn, n, An
340     print using "p#=#.#############"; n, P: print
350     Yn1=(1-sqr(1-pow(Yn,2#)))/(1+sqr(1-pow(Yn,2#)))
360     An1=pow(1+Yn1,2#)*An-pow(2#,n+1#)*Yn1
370     differ=abs(pi()-P)
380     P=1#/An1
390     Yn=Yn1: An=An1: n=n+1
400   until differ<lim
410   return()
420 endfunc
430 /* part 3
440 func pi_3()
450   float Yn, Yn1, An, An1, P, differ
460   int n=0
470   Yn=sqr(2#)-1: An=6#-4#*sqr(2#): P=1#/An
480   repeat
490   /*print using "  Y#=#.#############  A#=#.#############"; n, Yn, n, An
500     print using "p#=#.#############"; n, P: print
510     Yn1=(1-pow(1-pow(Yn,4#),1#/4#))/(1+pow(1-pow(Yn,4#),1#/4#))
520     An1=pow(1+Yn1,4#)*An-pow(2#,2#*n+3#)*Yn1*(1+Yn1+pow(Yn1,2#))
530     differ=abs(pi()-P)
540     P=1#/An1
550     Yn=Yn1: An=An1: n=n+1
560   until differ<lim
570   return()
580 endfunc
590 /*
```

ないような途方もないレベルになってこそ，この方法はパワーを発揮するのである。必要となる精度を得るまでループの中を繰り返し回るのだが，1回ループを回るごとにπの真の値と一致する桁数が2倍，または4倍と，まさに指数関数的な勢いで伸びていくのである。これが，逆正接関数の無限級数による方式だと，1項を計算して加算しても，精度はせいぜい2～4桁ずつしか上がっていかない。今年2億桁の記録を樹立したのは，表3（3）のアルゴリズムなのだが，そのループ回数はたったの14回だという。

ということは，2億桁の数をいくつか格納するだけのメモリと，2億桁の数の足し算，引き算，掛け算，割り算，それに平方根の算出を高速で行えるプログラムさえあれば，あとはリスト5に載せたような命令を並べるだけで，好きなだけπの値が求められる。そう，プログラムさえあれば。かの計算に使われたプログラムはいうまでもないが，2億桁などという現実とは遠く離れた数を計算するものだから（打ち出すとプリンタ用紙40箱必要），その計算量も手法も想像を絶するというほどではないにしても，並みのものではない。僕らが小学校で習ったような掛け算，割り算のやり方はわかりやすい半面，計算機のアルゴリズムとして使うには効率が悪すぎるのである。

そこで「現時点ではもっとも効率のいいやり方」というやつを調べてはみたのだが，スピードを極限まで追求するやり方だけにパソコンにはちと荷が重いようだ。結局そのやり方を実行することを僕は断念したのだ。なにしろ億（少なくとも万）の桁に達する計算である。はっきりいって，非常に難しいが，そのわりに，円周率を計算する以外にはほとんど使い道がない。それで，なんだかばかばかしくなってきた，というのもある。X68000のメモリがちょっとくらい広いといっても，10万桁計算できればいいほうであろう。その高速の掛け算には大量の浮動小数点演算が必要なので（61ページを参照），コプロセッサもなしに無謀な戦いを挑んで，かえって遅くなってしまった，というのではいただけない。実際，10万桁くらいだったら速いアルゴリズムを使って得する分よりも，浮動小数点演算をして損する分のほうが大きく響いてくるであろう。真面目に，地道に，小学生の真似をしたほうが結局は速いということもありうる。急がば回れ，まるでウサギとカメのお話のようだ。教訓めいてるなあ。

多桁演算への道程

このままではあんまりなので，せめてカメさんになって地道に計算するプログラムを作ってみた。まず最初にお断りしておくが，このプログラムは実用的かどうかという点ではまったく使いものにならない。多桁演算サブルーチンというよりは，筆算シミュレーションといったほうがいいくらいのプログラムの集まりにすぎない。まあ多倍精度の数を扱うための1ステップ，試み程度のものだと思っていただきたい。

まず，ここで扱っている数は，

1）10進数
2）正の数
3）固定小数点

という，かなり手抜きなものだ。このフォーマットのいけないのは，

1）コンピュータは10進数が苦手。16進数か，せめて2進数にしないと遅くてしようがない
2）正の数だけでなにができるの？ 小学生じゃないんだから
3）扱える数の範囲が事実上限定されてしまう。あまり大きな数は扱えない

といったところだろう。どこかいいところがあるかといえば，

1）わかりやすい

この一点に尽きる。わかりやすいというのは，どんなデメリットにも勝るメリットなのだ。もっともそれは，プログラマにとっ

8）僕は「天才」などという表現は滅多に使うものではないと思っているが，確かにこの世には，天才の名に恥じない，凡人には到底理解できないような人間がたまに現れるのだ。
9）かなり難しい理論だというのに，ラマヌジャンはそれをほとんど直感的に見抜いた。しかし見抜いた時点で十分に納得して考えるのをやめてしまい，それ以上の形式的な証明を一切していない。彼にとってその必要はなかったのだ。彼はまた，研究成果をノートに書き記したのだが，その中身にしても，ごく簡単な注釈が添えてあればまだいいほうで，ほとんどは結果の公式だけ，しかも自己流の表記をしていた。それらはのちの数学者たちをひどく悩ませ，ノートの解読と証明にはかなり手間取ったということだ。

リスト6　多桁π計算（反復法）

```
 1: /*                                  */
 2: /*        π の 計 算 ( 反 復 法 )     */
 3: /*                                  */
 4:
 5:
 6:
 7: #define        WORD        short              /* 4 桁 = 1 ワ ー ド        */
 8:
 9: #define        NUMB        32                 /* 桁 数 ( バ イ ト 単 位 )  */
10: #define        NUMW        NUMB/2             /* 桁 数 ( ワ ー ド 単 位 )  */
11:
12: #define        CARDINAL    10000              /* 基 数                    */
13:
14: #define        MULDEPTH    (NUMW/2+1)         /* 掛 け 算 の 深 さ         */
15: #define        DIVLENGTH   NUMW*2             /* 割 り 算 用 ワ ー ク の サ イ ズ */
16: #define        SQRLENGTH   NUMW*2             /* 平 方 根 用 ワ ー ク の サ イ ズ */
17: #define        DUM         8                  /* デ ー タ 保 護 の た め の 隙 間 */
18:
19:
20: void           add( WORD*, WORD*, WORD*, int );
21: void           sub( WORD*, WORD*, WORD*, int );
22: void           mul( WORD*, WORD*, WORD*, int );
23: void           mul1( WORD*, WORD*, int, int );
24: void           div( WORD*, WORD*, WORD*, int );
25: void           sqrt( WORD*, WORD*, int );
26: int            sqrt1( WORD );
27: void           wkclr( WORD*, int );
28: void           copy( WORD*, WORD*, int );
29: int            cmp( WORD*, WORD*, int );
30: void           prt( WORD*, int );
31:
32:
33: WORD           dummy0[DUM], mulwk[MULDEPTH*2], dummy1[DUM], divwk[DIVLENGTH],
34:                dummy2[DUM], sqrwk0[SQRLENGTH], dummy3[DUM], sqrwk1[SQRLENGTH],
35:                dummy4[DUM], sqrwk2[SQRLENGTH], dummy5[DUM], sqrwk3[SQRLENGTH],
36:                dummy6[DUM], sqrwk4[SQRLENGTH], dummy7[DUM];
37:
38:
39:
40:         WORD   d0[DUM], Ak0[NUMW*2], d1[DUM], Ak1[NUMW*2],
41:                d2[DUM], Yk0[NUMW*2], d3[DUM], Yk1[NUMW*2],
42:                d4[DUM], Wk0[NUMW*2], d5[DUM], Wk1[NUMW*2],
43:                d6[DUM], Wk2[NUMW*2], d7[DUM], Wk3[NUMW*2],
44:                d8[DUM], One[NUMW*2], d9[DUM], Six[NUMW*2],
45:                da[DUM], Rt2[NUMW*2], db[DUM], PI[NUMW*2],
46:                dc[DUM];
47: void    main()
48: {
49:         WORD    i, m=8;
50:
51:         wkclr( One, NUMW*2 );
52:         One[0]=1;
53:         wkclr( Six, NUMW*2 );
54:         Six[0]=6;
55:         wkclr( Wk0, NUMW*2 );
56:         Wk0[0]=2;
57:         sqrt( Rt2, Wk0, NUMW+2 );
58:
59:         wkclr( Ak0, NUMW*2 );
60:         wkclr( Yk0, NUMW*2 );
61:         wkclr( Wk0, NUMW*2 );
62:         wkclr( Wk1, NUMW*2 );
63:         wkclr( Wk2, NUMW*2 );
```

てだけのことだが。使う側はたぶんプログラマのワガママなんか聞いちゃくれないだろう。そう，使い勝手が悪いということは，どんなメリットにも勝るデメリットなのだから。

本題に戻るが，ここで扱う数は正確にいうと10進数ではない。むしろ1万進数だ。10進数を4桁ずつ区切って，それを1桁のようなものと見ている。この4桁を，以後「ワード」という単位で呼ぶことにしよう。

固定小数点と書いたが，整数部は1ワード。あとはみんな小数部。あまり大きな数は扱えない。うすうす気づいた人もいるだろうが，πの計算に必要な範囲(これはBASICプログラムですでに確かめてある）をカバーし，かつそれ以上は扱えないようになっている。ちょっと面倒なプログラムを作るときは，必要最小限しか作らないほうがいい。「どんな数にも通用するルーチンを作るんだ」などと意気込んだら，必ずハマる。事実，この手抜きプログラムでさえも僕はかなり悩まされた。

このプログラムでは，四則演算（加算，減算，乗算，除算）と平方根の計算を行うサブルーチンを使って，πを反復法アルゴリズムで計算している。3番目に紹介したやり方がいちばん効率がよさそうなので，それを使っている。

使用言語はC。アセンブラは使いたくないし，しかし速さもほしい，できれば高級言語が，となれば答えはひとつだが，理由はほかにもある。このプログラムでは，しばしばワークエリアからわざとはみ出して計算することがあり，そのときに，配列の添え字のチェックがあまい，というかチェックをしないCだと非常に助かる。ここではCの欠点を逆手に取っているわけで，非常に愉快である。

気になるスピードの点だが，この節の初めに書いたとおり，結論からいって使いものにならない。おそらく1万桁程度だったら，マシン語で書いたarctan( )を使うプログラムのほうが数百倍も速い。なかでも特に遅いのが平方根だ。かなり複雑な処理をしているので，バカみたいに遅い。やはり去年，反復法でπを出そうとした（こちらの試みは，現在中断状態）ときに作った，マシン語の平方根のプログラムがあるのだが，そいつの，ざっと100倍の時間がかかっている。いくらコンパイラとはいっても，やはりまだまだMPUの力を限界までは引っ張り出せないようだ。もっとも，限界まで引っ張り出したとしても，1万桁程度では，最適化したarctan( )専用のプログラムのス

```
 64:           wkclr( Wk3, NUMW*2 );
 65:           mul1( Wk1, Rt2, 4, NUMW+2 );
 66:           sub( Ak0, Six, Wk1, NUMW+2 );
 67:           sub( Yk0, Rt2, One, NUMW+2 );
 68:
 69:           i=0;
 70:           for (;;) {
 71:
 72: /*                  printf( "  a%.1d=", i ); prt( Ak0, NUMW );*/
 73: /*                  printf( "  y%.1d=", i ); prt( Yk0, NUMW );*/
 74:                     div( PI, One, Ak0, NUMW+2 );
 75:                     printf( "PI=" ); prt( PI, NUMW-1 );
 76:
 77:
 78:                     mul( Wk0, Yk0, Yk0, NUMW+2 );
 79:                     mul( Wk1, Wk0, Wk0, NUMW+2 );
 80:                     sub( Wk0, One, Wk1, NUMW );
 81:                     sqrt( Wk1, Wk0, NUMW+2 );
 82:                     sqrt( Wk0, Wk1, NUMW+2 );
 83:
 84:                     sub( Wk2, One, Wk0, NUMW );
 85:                     add( Wk1, One, Wk0, NUMW );
 86:
 87:                     div( Yk1, Wk2, Wk1, NUMW );
 88:
 89:
 90:                     add( Wk0, One, Yk1, NUMW );
 91:                     mul( Wk1, Wk0, Wk0, NUMW+2 );
 92:                     mul( Wk2, Wk1, Wk1, NUMW+2 );
 93:                     mul( Wk3, Ak0, Wk2, NUMW+2 );
 94:
 95:                     mul( Wk1, Wk0, Yk1, NUMW+2 );
 96:                     add( Wk2, Wk1, One, NUMW );
 97:                     mul( Wk1, Wk2, Yk1, NUMW+2 );
 98:                     mul1( Wk2, Wk1, m, NUMW+2 );
 99:                     m *= 4;
100:
101:
102:                     sub( Ak1, Wk3, Wk2, NUMW );
103:
104:                     copy( Ak0, Ak1, NUMW+2);
105:                     copy( Yk0, Yk1, NUMW+2);
106:
107:                     i++;
108:           }
109:
110:           return;
111: }
112:
113:
114:
115: /*        演算サブルーチン                  */
116: /*        固定小数点・正数専用              */
117:
118:
119: void      add( n, n1, n2, l )                                          /* 足し算                */
120: WORD      *n, *n1, *n2;                                                /* n=n1+n2               */
121: int       l;
122: {
123:           int      i, m, carry=0;
124:
125:           for ( i=l-1; i>=0; i-- ) {
126:                     m=n1[i]+n2[i]+carry;
127:                     if ( m>=CARDINAL ) {                               /* 桁上げ                */
128:                               carry=1;
129:                               m -= CARDINAL;
130:                     } else carry=0;
131:                     n[i]=m;
132:           }                                                            /* オーバーフローは無視  */
133:
134:           return;
135: }
136:
137:
138: void      sub( n, n1, n2, l )                                          /* 引き算                */
139: WORD      *n, *n1, *n2;                                                /* n=n1-n2 (但し n1>n2)  */
140: int       l;
141: {
142:           int      i, m, carry=0;
143:
144:           for ( i=l-1; i>=0; i-- ) {
145:                     m=n1[i]-n2[i]-carry;
146:                     if ( m<0 ) {                                       /* 桁下げ                */
147:                               carry=1;
148:                               m += CARDINAL;
149:                     } else carry=0;
150:                     n[i]=m;
151:           }
152:
153:           return;
154: }
155:
156:
157: void      mul( n, n1, n2, l )                                          /* 掛け算                */
158: WORD      *n, *n1, *n2;                                                /* n=n1*n2               */
159: int       l;
160: {
161:           int      i, m;
162:
163:           wkclr( n, l );                                               /* 下半分の桁は計算しない */
164:           for ( i=l-1; i>=0; i-- ) {
165:                     mul1( mulwk+1, n1, n2[i], l+2-i );
166:                     add( n+i-1, n+i-1, mulwk, l+3-i );
167:           }                                                            /* オーバーフローは無視  */
168:
169:           return;
170: }
171:
172:
173: void      mul1( n, n1, k, l )                                          /* 1桁の数との掛け算     */
174: WORD      *n, *n1;                                                     /* n=n1*k                */
175: int       k, l;
176: {
177:           int      i, m, carry=0;
```

ピードにはとてもじゃないが追いつけない。100万桁，1000万桁といったバカでかい数にならないと，つまりループをかけた効果が本当に表れるときまでは，おそらくもとは取れまい。この最低条件をクリアするのは，パソコンには荷が重いのだろう。

プログラムの解説はあまりする気がしない。足し算，引き算，掛け算はまあ当たり前の手順をふんでいるが，割り算や平方根は本当に面倒なので，今思い出すだけでもぞっとする。そうそう，計算桁数は，プリプロセッサのNUMBで指定する。10進で(NUMB×２)桁の計算をすることになる。

まあとにかく走らせてみよう。PI＝と書かれた後ろの数字が，しだいにπに近づいていくのがわかるだろうか。といっても，ふつうの人は，πの値をせいぜい５～６桁しか知らないだろうから，まるで無意味な数字に見えるかもしれない。πの値が変わらなくなったときに，おおむねいい精度で出ていると判断してほしい。というのは，掛け算や割り算をする桁数は限られているからだ。たとえばπを100桁求めようと思ったら，途中の計算で使う値の精度は，それより少し深く，たとえば104桁とか，108桁とか取らなくては，最後の桁あたりで誤差が出てしまう。

さっきカメさんなどといったが，実はタコさんだったようだ。もうちょっとましなプログラムを組みたかったのだが。まあそれでも基本的なところは教えたつもりだから，我と思わん方は10万桁でも100万桁でもやってほしい。

最後に

結局，円周率なんて，茶筒でも持ってきて，巻尺で測ればいいんだ，と逃げてこの話を終わることにしよう。完全な茶筒と正確な巻尺があれば，たった１回の割り算でπを求めることができるのだ。円周率はハマると本当に恐ろしい。僕は深入りしたくない。それにしても，今回はやたら小学生の昔を思い出したような気分だ。今だったらごく当たり前に，なかば無意識にできる計算も，あの頃はウンウンうなりながらやっていたのだ。そしてそれは，コンピュータにやらせても同じだった。

参考文献

金田康正，「円周率の計算ー世界記録を更新中」，UP174号，東京大学出版会

J. M. ボールウェイン/P. B. ボールウェイン，「天才数学者ラマヌジャンとπ」，サイエンス，1988. 4.


```
178:
179:         for ( i=l-1; i>=-1; i-- ) {
180:                 m=n1[i]*k+carry;
181:                 if ( m>=CARDINAL ) {
182:                         carry=m/CARDINAL;
183:                         m %= CARDINAL;
184:                 } else carry=0;
185:                 n[i]=m;
186:         }                                               /* オーバーフローは無視   */
187:
188:         return;
189: }
190:
191:
192: void    div( n, n1, n2, l )                             /* 割り算                 */
193: WORD    *n, *n1, *n2;                                   /* n=n1/n2                */
194: int     l;
195: {
196:         int     i, i1, l1, m;
197:         double  m1, m2;
198:
199:         copy( divwk, n1, l );
200:         n1=divwk;
201:         l1=l+1;
202:         for (;;) {                                      /* 頭に０が並ぶ場合は     */
203:                 if ( n1[0]==0 ) {                       /* そこを飛ばす           */
204:                         n[0]=0;
205:                         n++;
206:                         n1++;
207:                         l1--;
208:                 } else break;
209:         }
210:         for (;;) {
211:                 if ( n2[0]==0 ) {
212:                         n--;
213:                         n2++;
214:                 } else break;
215:         }
216:         m2=(double)n2[0]*CARDINAL+(double)n2[1];
217:         i=0;                                            /* 頭2ワードで商を近似する*/
218:         m1=(double)n1[0]*CARDINAL+(double)n1[1];
219:         for ( i=0; i<l1; i++ ) {
220:                 m=m1/m2-1;
221:                 if ( m>0 ) {                            /* 商が０でなければ引く   */
222:                         mul1( mulwk+1, n2, m, l1 );
223:                         sub( n1+i-1, n1+i-1, mulwk, l1-i+1 );
224:                 }
225:                 if ( m>=0 ) {                           /* もう１回引けるか？     */
226:                         if ( cmp( n1+i-1, n2-1, l1-i+1 )>=0 ) {
227:                                 sub( n1+i-1, n1+i-1, n2-1, l1-i+1 );
228:                                 m++;
229:                         }
230:                         n[i]=m;
231:                 }                                       /* ２回目からは頭３ワード */
232:                 m1=((double)n1[i]*CARDINAL+(double)n1[i+1])*CARDINAL+(double)n1[i+2];
233:         }
234:
235:         return;
236: }
237:
238:
239: void    sqrt( n, n1, l )                                /* 平方根（開平法）       */
240: WORD    *n, *n1;
241: int     l;
242: {
243:         int     m, i, falsemin, truemax;
244:
245:         wkclr( sqrwk0, l*2 );
246:         wkclr( sqrwk2, l );
247:         wkclr( sqrwk4, l );
248:         m=sqrt1( n1[0] );                               /* 整数部をとる           */
249:         sqrwk0[0] = n1[0]-m*m;
250:         n[0]=m;
251:         sqrwk2[0]=m*2;
252:         for ( i=1; i<l; i++ ) {
253:                 sqrwk0[i*2-1]=n1[i*2-1];                /* ２桁ずつ持ってきて     */
254:                 sqrwk0[i*2]=n1[i*2];                    /* 1桁ずつ出す            */
255:                 copy( sqrwk1, sqrwk0, i*2+1 );
256:                 if ( i==1 && m==0 ) falsemin=9999; else {
257:                         div( sqrwk3, sqrwk1+i-1, sqrwk2-1, i+2 );
258:                         falsemin=sqrwk3[0]+1;           /* 近似値（上限）         */
259:                 }
260:                 sqrwk4[i-1]=1;
261:                 add( sqrwk2, sqrwk2, sqrwk4, i+2 );
262:                 div( sqrwk3, sqrwk1+i-1, sqrwk2-1, i+2 );
263:                 truemax=sqrwk3[0];                      /* 近似値（下限）         */
264:                 sub( sqrwk2, sqrwk2, sqrwk4, i+2 );
265:                 sqrwk4[i-1]=0;
266:                 while ( falsemin>(truemax+1) ) {        /* 上限と下限の間を       */
267:                         m=(falsemin+truemax)/2;         /* 繰り返しとる           */
268:                         sqrwk2[i]=m;
269:                         mul1( sqrwk3, sqrwk2-1, m, i+2 );
270:                         if ( cmp( sqrwk0+i-1, sqrwk3, i+2 )<0 )
271:                                 falsemin=m;
272:                           else  truemax=m;
273:                 }
274:                 m=truemax;
275:                 sqrwk2[i]=m;
276:                 mul1( sqrwk3, sqrwk2-1, m, i+2 );
277:                 sub( sqrwk0+i-2, sqrwk0+i-2, sqrwk3-1, i+3 );
278:                 n[i]=m;
279:                 sqrwk4[i]=m;
280:                 add( sqrwk2, sqrwk2, sqrwk4, i+1 );
281:                 sqrwk4[i]=0;
282:         }
283:
284:         return;
285: }
286:
287:
288: int     sqrt1( n )                                      /* 整数の平方根           */
289: WORD    n;
290: {
291:         int     i=1, j=1;
```

乱数シミュレーションとπ

本文では紹介しなかったが，ほかにもπを出すやり方はたくさんある。それだけπにとりつかれた人が多いということであろう。ここでは，そのなかでもいちばん有名なモンテカルロ法(Monte Carlo method）を紹介しよう。

モンテカルロ法は，ご存じフォン・ノイマン(J.von Neumann）らが提唱した方法で，

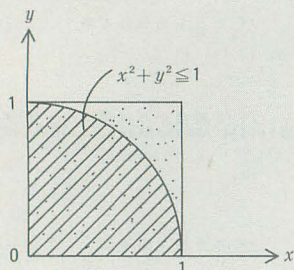
「決定論的な数学的問題の解決に乱数を用いること」

と定義づけられている。

これはコンピュータなしにはまず考えられない方法である。πの計算を例に取ってみよう。

図3のような4分の1円を考える。四角い範囲の中に，乱数を使って，点を山ほど打つ。すると，円の中に入った点の数は，理論的には4分の1円の面積，つまりπ/4になる。少なくともそれに近づく。

あまり好きなやり方ではないし，バカバカしいほど簡単だったのでプログラムはしなかったが，やりたい人はやってみるといい。はっきりいって，これは本文で紹介したどんな方法よりも効率が悪い。正確な乱数の扱いにくさについては華門氏の原稿を見てほしい。

図3 モンテカルロ法

$x=rnd()$

$y=rnd()$

$x^2+y^2 \leqq 1$ なら円の中

$\frac{(円内の点の数)}{(打った点の数)}$ は $\frac{\pi}{4}$ に近づく。

```
292:
293:            if ( n==0 ) return(0);                          /* Σ(2n-1)=n^2 を利用 */
294:            for (;;) {
295:                    n -= i;
296:                    i += 2;
297:                    if ( i>n ) break;
298:                    j++;
299:            }
300:            return(j);
301: }
302:
303:
304: void       wkclr( n, l )                                   /* ワークエリアをクリア */
305: WORD       *n;
306: int        l;
307: {
308:            int     i;
309:
310:            for ( i=0; i<l; i++ ) n[i]=0;
311:
312:            return;
313: }
314:
315:
316: void       copy( n, n1, l )                                /* コピー             */
317: WORD       *n, *n1;
318: int        l;
319: {
320:            int     i;
321:
322:            for ( i=0; i<l; i++ ) n[i]=n1[i];
323:
324:            return;
325: }
326:
327:
328: int        cmp( n1, n2, l )                                /* 大小比較           */
329: WORD       *n1, *n2;                                       /* sign(n1-n2)       */
330: int        l;
331: {
332:            int     i, s=0;
333:
334:            for ( i=0; i<l; i++ ) {
335:                    if ( n1[i]>n2[i] ) {
336:                            s=1;
337:                            break;
338:                    }
339:                    if ( n1[i]<n2[i] ) {
340:                            s=-1;
341:                            break;
342:                    }
343:            }
344:
345:            return( s );
346: }
347:
348:
349: void       prt( n1, l )
350: WORD       *n1;
351: int        l;
352: {
353:            int     i;
354:
355:            printf( "%.4d.", n1[0] );
356:            for ( i=1; i<l; i++ )
357:                    printf( "%.4d ",n1[i] );
358:            printf( "¥n" );
359:
360:            return;
361: }
```

大型機の掛け算方法

音声サンプリングをする人なら聞いたことのある言葉かもしれないが，フーリエ変換の応用である。概念だけでも説明しておこう。

n桁の2つの数を掛けあわせる場合を考えてみよう。

$a_{2n}\,a_{2n-1}\cdots a_{n+2}\,a_{n+1}\,a_n\,a_{n-1}\cdots a_2\,a_1$

$b_{2n}\,b_{2n-1}\cdots b_{n+2}\,b_{n+1}\,b_n\,b_{n-1}\cdots b_2\,b_1$

掛け算をすると桁数が2倍になることを考えて，上からn桁が0になっている2桁の数を用意しよう。

その掛けようとする数の1桁1桁を，別々のデータと見なして，フーリエ変換を掛ける。すると，次のような数列が2つ得られる。

$a'_{2n}, a'_{2n-1}, \cdots, a'_{n+2}, a'_{n+1}, a'_n, a'_{n-1}, \cdots, a'_2, a'_1$

$b'_{2n}, b'_{2n-1}, \cdots, b'_{n+2}, b'_{n+1}, b'_n, b'_{n-1}, \cdots, b'_2, b'_1$

掛け算には，同じ添え字のついた項どうしを掛ければよい。すなわち，

$c'_{2n}=a'_{2n}\times b'_{2n}$

$c'_{2n-1}=a'_{2n-1}\times b'_{2n-1}$

$\vdots$

というぐあいにやればよい。こうして次の数列を得る。

$c'_{2n}, c'_{2n-1}, \cdots, c'_{n+2}, c'_{n+1}, c'_n, c'_{n-1}, \cdots, c'_2, c'_1$

最後にこの数列をフーリエ逆変換してやる。ちょっとそのデータをいじれば，

$c_{2n}c_{2n-1}\cdots c_{n+2}\,c_{n+1}\,c_n\,c_{n-1}\cdots c_2\,c_1$

が出てくる。これが求める積である。

気づいている人もいるだろうが，「フーリエ変換をかける」だとか，「フーリエ逆変換してやる」だとか，「ちょっとデータをいじれば」だとか，曖昧な書き方をしているところがある。ここが浮動小数点の演算をしこたま使うところで，僕はここで投げ出したのだ（申し訳ない）。

さて，このやり方のどこが有利かというと，掛け算が2n回ですんでいるところである。実際はフーリエ変換でも少し掛け算を使っているが，それでもふつうの掛け算に比べ，はるかに手間は少ない。ふつうの掛け算はnの2乗に比例した数の手間がかかる。フーリエ変換を使えば，nに比例するだけですむ。もちろん，変換や逆変換にかかる時間を考えると，桁数が少ないうちはかえって遅いが，nが万だとか億といったオーダーに達すると，極端に速くなるのは想像にかたくない。誰だって，1度や2度，次のような計算をしたことはないだろうか。足し算と同じつもりで，縦に掛けるのである。昔は，どうしてこれが間違っているのか不思議に思ったが。

```
誤)  132            正)   132
   ×)534              ×)534
     598                  528
                         396
                        660
                        70488
```

これですんだら本当にいいのに，と思いながら一所懸命に右のようなやり方を練習したのだろう。フーリエ変換を使えば，ちょうど左のようなカンジになるのである。かなり重宝だと思うのだが。

超応用グラフィック　歪められた光

レイトレーシングでも数値演算は必修科目ですが，ここではちょっと道を踏み外して，空想の重力場が目の前にあったらと考えてみましょう。曲がる光をシミュレートできるなんてコンピュータグラフィックならではの楽しみだと思いませんか？

Kuwano Masahiko
桒野 雅彦

曲がる光を見てみたい

それは，特にすることもなくぼんやりと電脳倶楽部のユーティリティで遊んでいたとき頭のなかに浮かんだ疑問から始まった。「物体の周りでは空間が歪んでいるため，光も曲がって進むと，どこかで聞いたことがある。それなら思いっ切り重い星のそばを通る光はとんでもなく曲がるのだろう。極端な話，ブラックホールがあると，その周囲では恐ろしいほどねじ曲げられるはずだ。あの壁のポスター（エリア88だったりする）と俺の間に超コンパクトのブラックホール（そんなものがあるかどうかは別にして）を置いたとすると，きっと，とんでもなく歪んで見えるはずだ。……」

ここで，電脳倶楽部とはサヨナラして，原稿用紙と鉛筆とナイフ，コンパス，そして手持ちの物理っぽい本を引っ張り出す。確か，空間が歪むとか光が曲がって進むなんていうのは一般相対性理論の本では列車の時計が遅れるのと同じくらいポピュラーな話題として取り上げられているはず。曲がり方の式くらいはどこかに載っているだろうと，あちこちページをめくってみた。

残念ながら，手にした本のどれもが重力の影響で光が曲がることや，1919年の日食時に太陽のそばに見える星の位置のずれが，一般相対性理論から導かれる結果とよく一致したということには触れているのだが，それでは具体的に曲がり方がどのような式で表されるのかについては「この曲がりを計算する式は難解であるが……」くらいですっと逃げられてしまい，いきなり，

$$\theta=4GM/(RC^2)$$

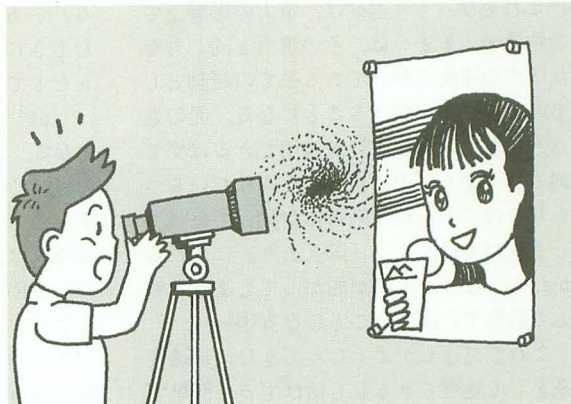
という式が出てきてしまうのである。

この式ではGは重力定数，Mは星の重さ，Cは光速度，Rは光がその星にもっとも接近したときの星の中心からの距離となっているのだが，考えてみると，重力は局所的に働くものではないのだから，光は一気にカックンと曲がるのではなく，グニュグニュっと曲がっていくはずである。それを，このような式で表しているということはこの式が，光は星のすぐそばにくるまでは直進すると仮定しているということにほかならない。

つまり，この式が成り立つのはごく近くまで来ないと相対論的な効果が現れない軽めの星の場合であり，ほんの少ししか曲がらない場合に限定されるのである。私が望んでいるのは思いっきり光が曲がる姿なのであるから，恐らく先ほどの半径に反比例するような曲がり方の式は適用できそうにない。試しに本屋もうろついてみたが，かんばしい成果は上がらなかった。

手元の本でも本屋で探した本でもはっきりしない，となれば自分で考えるより仕方がない。ニュートン力学にどっぷりと漬かった脳味噌をねじきれるほど絞りながらそこらじゅうの本をひっくり返して，一週間半。部屋じゅう式の計算用紙（Oh！Xの原稿用紙の裏だったりする）で埋めつくし，本当に足の踏み場もない状況にしてしまったあげく，一応それらしい式をでっちあげ，その勢いで無理やりプログラムを作ってしまった。

もともと相対論など聞きかじりであるうえ，周りに確認を求める人間もいないのでかなり強引で都合のよい仮定を積み重ねてしまっている。もともと，曲がる光をシミュレートしたいという欲求からでっちあげたものなので，実際の相対性理論とは掛け離れたものとなっているかもしれない。ちなみに，これを読んだ編集のU氏には，まだまだニュートン的な概念に支配された仮説だと言われてしまったほどである。

それでも一応，光がねじ曲がる感じは出せたし，出力結果がこれまた予想以上に興味深いものになったので，ここで特集の応用例としてすべり込ませてもらえることになったわけだ。現れてくる図形を眺めながら頭を抱え，首の筋を違えるくらいひねるような楽しみ方なら十分にできるだろうと思う。

言語としてはX-BASICを使ったが，たかだか70行程度のプログラムであるから他の言語で書き直すこともそれほど難しくはないだろう。

SFノリで仮説を作る

なんともよくわからないことだらけの相対論で，重力による空間の歪みを考えていくと，やれ時間の進み方が変わるだの，ローレンツ短縮が起こるだのとごちゃごちゃといろいろなことが混ざってきてわけがわからなくなる。私もこれにはまってしまい，まるまる一週間もの間頭を抱え込んでしまった。

で，あれこれと突っついたあげく，たどりついたのが「等価原理」である。そういえばそんなものがあったっけと気づくのが相当に遅くなってしまった。等価原理というのは慣性系の間で考えられていた相対性原理を加速度系をも含めた系の間で成り立つように拡張するために持ち込まれたものである。

「等価原理」などといかめしい字面であるがその考え方自体はごく単純である。加速度系にある物体には，その質量と系の加速度とに応じた慣性力が働く。これを系の加速度のせいであると考えるのではなく，系自体は慣性系なのであるが，その外側に重力を及ぼすような星があると考えてもかまわないだろうということなのである。

電車に乗ったとき，発車や停車の際に私たちは横方向に結構な力を感じるが，この力を進行方向と平行に働く重力と区別することはできない。また，自由落下する飛行機の中では無重力状態を体験できる。飛行機自体に注目すればこれはまぎれもなく加速度系であり，地球の重力はもちろん働いてはいるのであるが，その中にあるものにとってはなんらかの原因で重力が消し去られてしまったと考えることができる。

これをひっくり返せば，重力の影響下での物体のふるまいは，その重力と同じ力を及ぼす加速度で動いている系での運動として考えることができることになる。光の進み方にもこの等価原理が適用できるはずである。つまり，光は真っ直ぐ進んでいるつもりなのであるが，系自体が動いていってしまうために，光は「置き去り」にされるかたちとなり，進路が曲がってしまうと考えて計算すればよいことになる(図1)。

これで見通しがよくなってきた。加速度系として処理できるなら話はずっと簡単である。まず，星の中心からの距離と，その場所での重力の強さの関係を知る必要がある。このあたりについても，あまり記述がないのであるが，どこを眺めても，

$$f=GMm/R^2$$

Gは重力定数，Mは星の，mはその物体の質量，Rは星の中心からの距離

という恒例の，距離の2乗に反比例する式をそのまま適用している例しか見当たらないところを見ると，これをそのまま拝借してかまわなさそうである。

ここで，

$$f=ma \quad (aは加速度)$$

といういつもの式を代入することで，

$$a=GM/R^2$$

となる。つまり，星の中心からの距離Rの位置における物体のふるまいは，GM/R^2の加速度が系に働いていると考えて計算すればよいことになる。

力まかせに問題を解く

ここまでわかれば，あとは幾何学で攻めていけばよいだろう。

図2に示すように，中心から半径Rのところで，半径と角度ϕで交わる直線上を光が進んできたとき，この光がΔxだけ進むのにかかる時間は$\Delta x/C$(Cは光速度)である。一方，加速度aで時間tだけ加速されたときに進む距離は(初速度は0とする)，aをtで2回積分すればよいから，

$$at^2/2$$

となる。ここで，aにGM/R^2を，tに先ほどの$\Delta x/C$を代入し，系が進んでしまう距離，すなわち光が星の側に落ち込む距離ΔRを求めれば，

$$\Delta R=(GM/R^2)(\Delta x/C)^2/2$$

となる。

さて，ここで光はどの程度曲がって進むのであろう。当初の方向にΔxだけ進む間に中心方向にΔRだけ引きずりこまれることになるので，進路は図の平行四辺形の対角線の向きになる。

一方，進む距離は通常の物体なら「加速」されることになるので，進む距離はまさに対角線の長さそのものになるのであるが，ここで相手にしているのは光であるから，光速度一定の法則が適用される。いくら引きずりこまれてはいても$\Delta x/C$の時間ではΔxしか進むことはできない。これと，今度は平行四辺形を半分にした三角形から，このときのθについては，

$$\tan\theta=\Delta R\cos\phi/(\Delta R\cos\phi+\Delta x)$$

が成立することがわかる。現在のBASICにはtanの逆関数としてatanが用意されているのが普通なので，ここから即座にθを求めることができる。

これで，光の曲がる角度は計算できた。それではこの方向にΔxだけ進めることにしよう。さて次の計算のために，新しい進入角ϕと，中心からの距離Rを求めておかなくてはならない。これには，先ほどの平行四辺形に半分かかるような形になっている三角形(RとR'を二辺としているもの)を利用するとうまくいきそうである。

まず，新しいR(R'としておく)を求めてしまおう。最初のRを底辺と見たときの頂点から図のように垂線をおろして，図の左半分について三平方の定理を適用すれば，

$$R'^2=(R-\Delta x\cos(\phi-\theta))^2+(\Delta x\cos(\phi-\theta))^2$$
$$=R^2-2R\cdot\Delta x\cos(\phi-\theta)+\Delta x^2$$

R，R'はいつでも正の数だから，ここで両辺の平方根をとって，R'を求めることができる。

つぎに新しいϕを求めるわけだが，その下準備として図の円の中心でRとR'のなしている角，γを求めよう。ここに先ほど使った直角三角形を当てはめれば，次の式が成り立つ。

$$R'\cos(\gamma)=\Delta x\cos(\phi-\theta)$$

ここから，

$$\cos(\gamma)=\Delta x\cos(\phi-\theta)/R'$$

さて，ここでcosの逆関数ATANに習うならASINのようなものがあれば，いっぺんにγが求められるのだが，残念ながらX-BASICにはATANしかないので，ASINをATANで表さなくてはならない。

ここで，

$$\tan\theta=\sin\theta/\cos\theta$$
$$=\sin\theta/\sqrt{(1-\sin^2\theta)}$$

から，

$$ASIN(X)=ATAN(X/SQR(1-X^2))$$

となる。

これでようやくγを求めることができた。ここまでくれば，当初の目的である，次の計算で使うϕを求めることは簡単な角の和と差で求めることができるのである。

図1　加速度系と重力の影響の対比

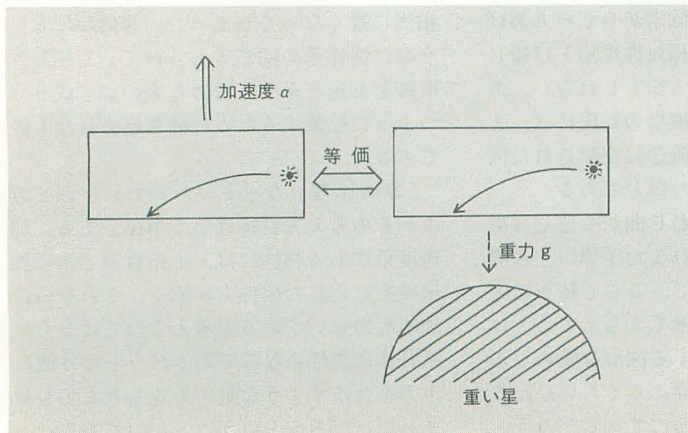
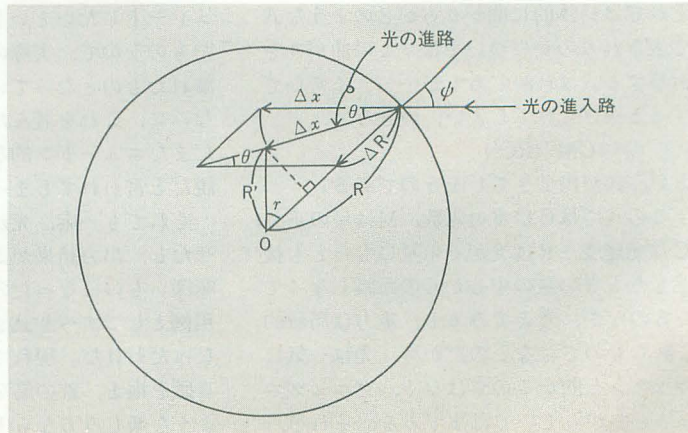



図2　進路計算

$\psi' = \psi - \theta + \gamma$

ちなみに新しいRは先ほど行った計算結果から，

$R' = \sqrt{(R^2 - 2R \cdot \varDelta x \cos(\psi - \theta) + \varDelta x^2)}$

となる。

エイヤッでプログラミング

ここまでできれば，プログラムを組んでみることができる。$\varDelta x$やθ，Mなどの値を適当に決めて，あとはジクジクとコンピュータに計算させ続ければ，進路を表示させることができる。私の作ったプログラム（リスト1）では，前方に，タイルのような正方形の集まった図形を，その手前にブラックホールをおいて，それを望遠鏡で覗いたような状態を計算させてみることにした。

走らせると，まず画面の真ん中，ブラックホールを置いたところを中心とする円を描いたあと，右側（視点）から出た光の進みぐあいを計算しながら，その結果が表示される。左の端に，タイルが並んでいると考えてもらいたい。

ひと通り計算が終わると，いよいよ，どのように見えるかの表示に移ることになる。数値演算が多いのでかなり遅いがじっと我慢の子でいるしかない。

出てきた結果はいかがであったろうか？私は初めて結果を見たとき，思いっ切り首をかしげてしまったのであるが……。

プログラムはもはや解説するほどのものではないだろう。ちょっと触れておくと，このプログラムでは$GM/2C^2$をKという変数で表しており，この値が大きいほど，大袈裟な曲がり方をする。コンピュータであるからいくらでも大きな値にすることが可能であるが，面白いのはせいぜい10から20くらいまでで，それ以上になると曲がり方が激しすぎて，わけがわからないだけのものになってしまうようである。

ではまた

これでめでたく完成となったのであるが，なにぶんにも私の勝手な思い込みと，想像ででっち上げた式に基づいているので，ひょっとしたら，というより，ほぼ間違いなくとんでもないチョンボをしでかしているような気がしなくもない。

さらに付け足すと，本当はブラックホールの近くを通った光は大きな赤方変移を受けるはずだと思うのだが，それをどのようにカラー・パレットコードに変換したらよいのか考え切れなかったので，今回のプログラムではここまで対応し切れていない。

まあ，それでも光が曲がって進むと，どんなことになるかということを目にすることができたのはなかなか感動的なものがあった。もし，読者のなかで，「相対論ならまかせておけ」という方がおられたら，ぜひまともな理論ではどうなっているか教えていただきたい。

図3　タイルパターンの見え方

1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

16色に塗り分けられたタイルパターン

↓

リスト1　曲がった光を見るプログラム

```
1000 screen 2,0,1,1
1010   circle(300,200,10,15)
1020 int i,j,x,y
1030 float r
1040 dim int taget(200)
1050 float py
1060 str s
1070 for i=0 to 199
1080         py=bh_cal(atan(i/600#))
1090         if py>=0 then {
1100                 taget(i)=399-py
1110         } else  taget(i)=-8000
1120 next
1130 wipe():cls
1140 circle(200,200,200,15)
1150 for j=0 to 399
1160     for i=0 to 399
1170         r=sqr((i-200)*(i-200)+(j-200)*(j-200))
1180         if r > 199 then continue
1190         if taget(r)=-8000 then continue
1200         x=taget(r)*1#/r*(i-200)+200
1210         y=taget(r)*1#/r*(j-200)+200
1220         pset(i,j,get_col(x,y))
1230     next
1240 next
1250 img_save("BLACKS")
1260 print"ワタシハ・・・";:input s
1270 end
1280 /**/
1290 func get_col(x;int,y;int)
1300         if x<0 or x>399 or y<0 or y>399 then return(0)
1310         x=x ¥ 100
1320         y=y ¥ 100
1330         return(((y*4+x) mod 15)+1)
1340 endfunc
1350 /**/
1360 func float bh_cal(P;float)
1370     int   i,px,py,nx,ny
1380     int   DX
1390     float R,K,DR,RD,tmp
1400     float T,G
1410     float anglesum
1420     /**/
1430     anglesum=0
1440     DX=5
1450     K=10#
1460     R=300#
1470     px=R+300:py=200
1480     for i=0 to 150
1490         DR=K*DX*DX/(R*R)
1500         T=atan(DR*sin(P)/(DX+DR*cos(P)))
1510         RD=sqr(R*R-2*R*DX*cos(P-T)+DX*DX)
1520         if RD < 10 then ny=-1:break
1530         tmp=DX*sin(P-T)/RD
1540         G=atan(tmp/sqr(1-tmp*tmp))
1550         if tmp<0 then G=-G
1560         /****/
1570         P=P-T+G
1580         anglesum=anglesum+G
1590         R=RD
1600         nx=cos(anglesum)*RD+300:ny=sin(anglesum)*RD+200
1610         if nx<0 then ny=(px*ny-py*nx)/(px-nx):line(px,py,nx,ny,15):break
1620         if nx>600 or ny<0 or ny>400 then ny=-1:break
1630         line(px,py,nx,ny,15)
1640         px=nx:py=ny
1650     next
1660     if (i>150) then ny=-1
1670     return(ny)
1680 endfunc
```

特集 真夏の夜の数値演算

超応用AD PCM 音の数学

自然の音をそのまま再生してくれるAD PCM。これをうまく使えば，いろいろな音が合成できるのでは？「音は連続関数だ。連続関数ならなんとかなる」こうして，フーリエ級数，微積分による究極の音声合成への挑戦が始まったのです。

Katou Masaya
加藤 賢哉

音を表す

はるかな昔，人間に許された情報交換の手段は音声であり，人の音声をそのままに伝えることができたのは，その人の声が届く範囲内だけだった。ある程度の文明が現れると，太鼓などの人間の声よりも大きな音を発生するものや，煙などの視覚に訴えるものにより，かなり遠方との間で情報交換が可能となった。文字が発明されるとより細かい意思疎通ができるようになったが，それはかなり時間を要する方法に頼るしかなく，ごく近代まではリアルタイムな情報交換の手段としては適当ではなかった。このような状態に新風を吹き込んだのは電気(動電気)の実用化である。これにより，有線電信が可能となった。そして，電波の発見により無線電信が実用化されたのだ。

さて，音声は空中や電線の中をどのように伝わっているのだろうか。そのままのかたちで伝わっていないことは電話線に耳をあてるまでもなく明らかだ。

音声の変調方法でよく知られているのが次の2つだろう。すなわち，AM（振幅変調）とFM（周波数変調）だ。これらは放送電波のフォーマットとして有名だが，ここで取り上げるのはパルス変調という方法である。というのも，AD PCM という方式はパルス変調の一種なのだ。

パルス変調には，

1) PAM
2) PCM
3) PWM
4) PDM

などがある。これらについてざっと解説しよう。

PAMはパルス振幅変調であるが，私はこれがどのようなものに使われているか知らないので特に説明はしない。信号系のエンハンスが楽にできるのでビデオ機器などで使われることもあるらしい。

PCMはパルスコード変調で情報を符号化して伝送する方法である。モールス信号などが代表的なものであるが，コード化と共にエラーチェックなどを加えることで高い信頼性を持たせることもできる。情報機器の通信などに使われることが多い。

PWM はパルスの幅に情報を持たせるものであり，シャープ系のマシンのデータレコーダに採用されている伝送方法だ。これは速度変化に強い（追従性に富む）という特徴を持つ。よってシャープ系のデータレコーダでは高いボーレートにもかかわらず，データエラーがほとんどないのだ。

PDM はパルスの密度で信号を送るもので，すべての動物の神経系で行われている情報伝達方式だ。神経を伝わる信号はパルス密度の変化によって表される。最近では脳に電極を取り付けて怒りたくないのに怒ったり，なにもしないのに激痛を与えたりすることができるそうである。

AD PCMとはなにか

というわけで，問題のPCMであるが，もっとも身近なPCMはなにか？ まず，一部のビデオにはPCM変調による音声記録方式が採用されている。CDプレイヤーはPCM復調機といえる。そして，DATは明らかにPCM変/復調機である。それではこのなかでもっとも普及していると思われるCDプレイヤーを例にPCMというものを見てみよう。

CDの基本フォーマットは量子化16ビット，サンプリング周波数44.1kHzである。これは1秒間に44100回音の大きさを調べ，それを65536段階に分けて表現するということを意味している。

さて，音の高さはどこへいってしまったのかと思う人もいるかもしれない。基準音のA（ラの音）は440Hzだそうだ。これが仮にサイン波だとすると，∾のような波形が1秒間に440回あるということである。1オクターブ上のラの音は880Hzだ。当然∾が1秒間に880回あることになる。音というのは基本的に周期関数で表すことができるので，すべての音階は1秒間に何回その波形が現れるかで決定できることになる。そして，ひとつの波形は音量の変化として表すことができる。というわけで，音の高さのパラメータはなくてもいいのだ。音色なども波形の変化によって表されるので，特別なパラメータは一切必要ない。PCMでは音量の変化だけですべてが決定される。

ふつうのCDは1枚に74分入るので，

$$74 \times 60 \times 44100 \times 16 \div 8 = 391608000$$

すなわち，400Mバイト近い容量を持つことになる。最近の高級機の標準は18ビット8倍オーバーサンプリング（352.8kHz）だから，1.7Gバイトものデータ量となる。はっきりいって，これは容易に扱える大きさではない。もっと効率のよい方法はないのだろうか？

音声には基本的に連続性がある。すなわち，前後のデータに関連があるといえる。そこで登場するのが⊿PCM方式だ。⊿はデルタと読み，差分を表す場合に多用される記号である。

たとえばここに，

2,3,5,7,11,13,17,19,23

という数列があったとする。これをそのままPCMで保存するためには，9個×5ビット（データの範囲），つまり最低45ビット必要である。しかし，これを，

2,1,2,2,4,2,4,2,4

という差分に置き換え，それぞれから1を引いてしまうと，9個×2ビット，すなわち18ビットと半分以下の容量ですむ。

CDの場合，音の大きさを0から最大値までの65536段階に分けているのだが，これが1/44100秒の間に全体の1/256も変化することがあるだろうか？ これは音が1秒間に127回，最低値から最大値まで変化できるほどの変化量である。ほとんどの場合(特に音楽でなく音声の場合)，8ビット，44.1kHzの⊿PCMでも十分な音質を得ることができる。これでメモリの使用量は半分になった。

それでは欠点はないのかというと，もちろんある。それは勾配過負荷といって変化量が大きすぎてその⊿PCMの差分の範囲で

は信号の変化に追従できない場合もありうるということだ。

さて，困った。なるべく過負荷を起こさないようにできないものか。そこで考案されたものがD PCM方式である。DとはDifferentialの頭文字だ。⊿PCMでは差分は前または後ろのデータとの差分であった。DPCM方式では，差分は過去のデータから予測された値と実際に入力された値との差をその差分値とする。これで勾配過負荷の発生はぐんと少なくなり，差分のビット数も少し減らすことができるようになった。

しかし，これでも勾配過負荷はまったくななくなったわけではない。そこで，ある人がいいことを思いついた。3ビットのDPCMでは，

+3，+2，+1，0，−1，−2，−3

という7通りの差分しか表せない。もしも4以上の差分がきたら勾配過負荷が発生してデータが不正確になってしまう。それでは4以上の差分が発生したときはきざみを倍にして，

+6，+4，+2，0，−2，−4，−6

として処理してしまおう。うーん，Adaptiveだ。といったかどうかは知らないが，この方式を使えば勾配過負荷の発生は疑似的に0になる。しかし多少の不正確さは残る。これがAD PCM方式である。

さて，X68000のAD PCMであるが，困ったことに⊿PCMらしい。というか⊿PCM的なデータを送るとそのとおりに動いてしまうのだ。どうやら，線形予測器が作動していない(と思う)のだが，どうしたものだろう。今回のサンプルも強引に⊿PCM用のデータに対応するものと仮定して作成されているが，一応うまく動いている。内部で線形予測器が動作していると振幅(音量)に誤差が出るが，周波数にはほとんど影響ないはずなのであまり問題はないだろう。

音を考える

さて，雑な説明だがAD PCMについてはある程度のイメージをつかんでもらえたと思う。ここで改めて，音とはなにかを考えてみたい。

我々が音として認識しているものは空気などの振動である。音を直接目で見ることはできないが，マイクで拾ってオシロスコープなどを使うと音の波形を見ることもできる。図1のようなものは誰でも見たことがあるだろう。ここで横軸は時間であり，縦軸は振幅すなわち空気の粗密状態を表す。こういった図ではわかりやすく横波の形で表現されるが，実際には振動は縦波として伝播する。

図を見てもわかるとおり，ある時間には必ずあるひとつの対応する状態がある。よって，これは関数であるということができる。しかも値が離散的になることはなく，常に滑らかな変化量を持った周期関数として表される。これは音を数学的に扱ううえで重要なポイントである。

さて，こういった振動のもっとも基本となるのが単振動という運動だ。これは図2のようなばね仕掛けを想像してもらうとわかりやすい。左右のばねはまったく同じ強度の完全弾性体で，ついでに質量は0に等しい。当然，摩擦係数は0だ。こういった運動を式にすると，

$$f(t)=k\cdot\sin 2\pi t$$

となる。これをそのまま音にすると，「ぽー」という感じのサイン波となる。

サイン波合成

たとえば，サンプリングシンセサイザとして有名なフェアライトCMIなどでは，ある基音に対して32倍音(機種によっては256倍音）までのサイン波を自由なエンベロープをつけて足しあわせることができる。基本的に楽器の音では基音に対する倍音構成がその音色を決定しているといってよい。実際には倍音以外の音も混じってはいるが，その量はごく微量である。リスト1では32

図1　音の波形

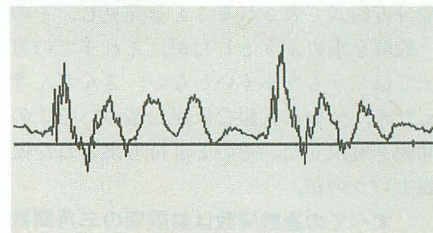
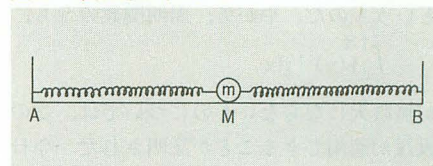
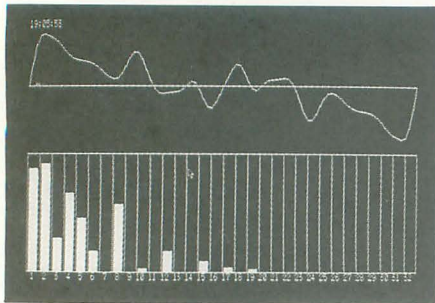



図2　単振動のイメージ

リスト1　サイン波合成

```
1000 int t,r,e,w,x,y,hz,rn,kz=1950,st=7
1010 float z,q,s,a0(99),a1(99),a2(99),n(32787)
1020 char a(65535)
1030 float p,nc=261.6255653#,pi=3.1415926535898#
1040 screen 2,0,1,1
1050 console ,,0
1060 mouse(0)
1070 mouse(1)
1080 mouse(4)
1090 msarea(1,257,766,478)
1100 str k
1110 box(0,256,767,479,15)
1120 for t=0 to 31
1130   line(t*24,256,t*24,479,15)
1140   locate -(t<9)+t*3,30
1150   print str$(t+1);
1160 next
1170 locate 0,0
1180 repeat
1190   mspos(x,y)
1200   msstat(w,w,t,r)
1210   if t=-1 then {
1220     x=x¥24
1230     fill(x*24+3,257,x*24+20,y,0)
1240     fill(x*24+3,y,x*24+20,478,13)
1250     a0(x)=(479#-y)/223#
1260   }
1270 until r=-1
1280 line(0,128,767,128,15)
1290 z=0
1300 for t=0 to 383
1310   q=0#
1320   for w=0 to 31:if a0(w)=0 then continue
1330     q=q+a0(w)*sin(pi*2##*t*(w+1)/768#)
1340   next
1350   if q>z then z=q
1360   t=t+15
1370 next
1380 q=0
1390 for t=st to 389
1400   p=q
1410   q=0
1420   for w=0 to 31:if a0(w)=0 then continue
1430     q=q+a0(w)*sin(pi*2*t*(w+1)/767)
1440   next
1450   line(t-st,128-p*100/z,t,128-q*100/z,13)
1460   line(767-t+st,128+p*100/z,767-t,128+q*100/z,13):t=t+st
1470 next
1480 print time$
1490 q=15600#/nc
1500 for t=1 to kz
1510   z=(t/q-int(t/q))*q
1520   p=0
1530   for r=0 to 31:if a0(r)=0 then continue
1540     p=p+a0(r)*sin(2*pi*z/q*(r+1))
1550   next
1560   n(t)=p
1570   if abs(p-n(t-1))>s then s=abs(p-n(t-1))
1580 next
1590 s=6/s
1600 for t=0 to kz/2-1
1610   e=int((n(t*2+1)-n(t*2))*s)
1620   if e<0 then e=8-e:if e>15 then e=15
1630   r=int((n(t*2+2)-n(t*2+1))*s)
1640   if r<0 then r=8-r:if r>15 then r=15
1650   a(t)=e*16+r
1660 next
1670 a_play(a,4,3,kz)
1680 print"a_play(a,4,3,";kz;")"
1690 print time$
1700 end
```

倍音までのサイン波を223段階で合成することができる。起動後マウスで各波形成分の割合を指定すると波形表示後，その音の合成して鳴らすというものだ。

なぜ，サイン波を足しあわせることで音を作ろうとしているのだろうか？　それは先ほど述べたとおり，音が連続かつ周期関数であるという事実に由来する。ざっと解説しよう。昔，フーリエという学者が金属板の1点を熱した場合の熱伝導が2階の偏微分方程式で表されることを発見し，その一般解を求めようとしたが，それまでの方法ではどうもうまくいかない。さんざん考えた末，ついに大胆な仮説に基づいてこの問題を解決した。そのとき持ち出された仮説というのが，

すべての連続関数は無限個の三角関数の和で表すことができる

というものだ。やがて，周期関数のうち，

$$\int_{-T/2}^{T/2} | f(x) | dx$$

が無限大にならないものについては，この仮説が適用できることが証明された。今日ではフーリエ級数は弦の振動や電流の状態などを表す偏微分方程式に関連して物理学，工学などで幅広く応用されている。ちなみに級数というのは似たような項がたくさんの足し算でつながった式のことだと思っておいてほしい。

すべての連続関数に対応するには，サイン波だけでなく，90°位相のずれた$\cos\theta$に相当する波の成分も必要なのだが，これは単一楽器の音色としては無視しておいてもよいだろう。

音声を変調する

リスト2と3だがリスト2はファイルより読み込んだサンプリングデータを任意の周波数にできるだけ正確に変調しようというプログラム，リスト3は任意の波形を手書きしてそれを任意の周波数のデータに落とすためのプログラムだ。リスト2では“bach.pcm”というファイル名を適当に変えて

リスト2　データを補間する

```
1000 int t,r=256,e,w,x,y,x1,y1,x2,y2,n(19999),hz=83
1010 float z,q,m(99),tt(99),xx(99),f(99),a0(99),a1(99),a2(99),a3(99)
1020 char a(7800)
1030 float p
1040 str k
1050 fread(a,7800,fopen("bach.pcm","r"))
1060 fcloseall()
1070 screen 2,0,1,1
1080 for t=1 to 383
1090   w=r
1100   e=a(t+0) shr 4
1110   if e>7 then e=8-e
1120   r=r+e
1130   line(t*2-1,w,t*2,r,15)
1140   n(t*2)=r
1150   w=r
1160   e=a(t) and 15
1170   if e>7 then e=8-e
1180   r=r+e
1190   line(t*2,w,t*2+1,r,15)
1200   n(t*2+1)=r
1210 next
1220 /*k=inkey$
1230 a_play(a,4,3)
1240 print"start point"
1250 repeat
1260   mspos(x,y)
1270   msstat(w,w,r,w)
1280 until r=-1
1290 x1=x
1300 y1=y
1310 print x,y
1320 for t=0 to 500
1330 next
1340 print"end point"
1350 repeat
1360   mspos(x,y)
1370   msstat(w,w,r,w)
1380 until r=-1
1390 x2=x
1400 y2=y
1410 print x,y
1420 r=x2-x1
1430 p=1#*(y2-y1)/r
1440 for t=x1 to x2
1450   n(t)=n(t)-p*(t-x1)
1460   pset(t,n(t),13)
1470 next
1480 screen 2,0,1,1
1490 console ,,0
1500 box(192,128,576,383,15)
1510 line(192,256,576,256,14)
1520 line(384,128,384,383,14)
1530 p=r/32#
1540 f(0)=256
1550 for t=0 to 32
1560   xx(t)=192+t*12
1570   f(t)=n(x1+p*t)
1580 next
1590 for t=0 to 32
1600   m(t)=(f(c(t+1))-f(t))/(xx(c(t+1))-xx(t))
1610 next
1620 for t=0 to 32
1630   r=abs(m(c(t+1))-m(t))*m(c(t-1))+abs(m(c(t-1))-m(c(t-2)))*m(t)
1640   e=(abs(m(c(t+1))-m(t))+abs(m(c(t-1))-m(c(t-2))))
1650   if e=0 then tt(t)=(m(c(t-1))+m(t))/2 else tt(t)=r/e
1660 next
1670 /*
1680 for t=0 to 31
1690   a0(t)=f(t)
1700   a1(t)=tt(t)
1710   a2(t)=(3*(f(c(t+1))-f(t))/(xx(c(t+1))-xx(t))-2*tt(t)-tt(c(t+1)))/(xx(c(t
+1))-xx(t))
1720   a3(t)=(tt(t)+tt(c(t+1))-2*(f(c(t+1))-f(t))/(xx(c(t+1))-xx(t)))/pow(xx(c(
t+1))-xx(t),2)
1730   for r=0 to 11
1740     pset(192+t*12+r,a0(t)+a1(t)*r+a2(t)*r*r+a3(t)*r*r*r,11)
1750   next
1760 next
1770 q=15600#/hz
1780 p=q/32:r=0
1790 for t=0 to 15599
1800   e=(t/p) mod 32
1810   z=t/p-int(t/p/32)*32
1820   z=(z-int(z))*12
1830   n(t)=a0(e)+a1(e)*z+a2(e)*z*z+a3(e)*z*z*z
1840 next
1850 for t=0 to 7800
1860   e=n(t*2+1)-n(t*2)
1870   if e<0 then e=8-e:if e>15 then e=15
1880   r=n(t*2+2)-n(t*2+1)
1890   if r<0 then r=8-r:if r>15 then r=15
1900   a(t)=e*16+r
1910 next
1920 a_play(a,4,3)
1930 print"a_play(a,4,3)
1940 end
1950 func c(t)
1960   if t=32 then return(t)
1970   t=(t+32) mod 32
1980   return(t)
1990 endfunc
```

表示したいデータを指定し，起動後はマウスで1波長分のデータを切り取ってほしい。リスト3の使い方は特に説明するまでもないだろう。

これらは要するに1波長分のデータを連続関数と見てそれに相当する関数を近似し，その関数から音長分のデータを算出しているわけだ。プログラム自体は非常にシンプルである。唯一わかりにくいと思われるところは，データの補間の部分だろう。

音声を変調する場合にもっとも単純な方法は与えられたデータを基準周波数のものと見たてて，そのデータを時間軸方向に定数倍するというものだ。この場合のデータは差分であるから，適当に詰めたり間引いたりして望みの周波数に近い波形を得るようにするわけだ。この方法では変調する周波数によっては，誤差が積もり積もって音色が変わったり，バラツキが出てくるなどの問題があり，どうもいまひとつ面白くない。そこでデータを補間してもっと奇麗な音を出してみようと思う。

補間といってもわからない人のほうが，多いと思われるので簡単に解説しよう。補間とは与えられた数個の点とその値より，求めたい点の値を計算することおよび，このための近似関数を求めることである。

補間法には全体をひとつの関数で補間する方式といくつかの小区間に分けて補間する方式がある。前者の方式として，

LAGRANGE
HERMITE
NEWTON
GAUSS

の方法などが有名である。なかでも，ラグランジュ（LAGRANGE）の方法は基本であり，簡単なものであるが，高次（4次以上）になると独特のゆれが生じて使いものにならない。そのほかの方法もラグランジュの方法と根本的には変わらないので，ここではこれらの方法は用いない。

さて，区分を分けて補間する方法（区分多項式）のうち，誰でも名前くらいは聞いたことがあると思われるのがSPLINEだ。しかし，SPLINE関数はときどき異常屈曲点が生じるのと，連立方程式を解かねばならないのでここでは用いない。応力SPLINE関数にすれば異常屈曲点はなくなるが，やはり行列式を解くのは面倒だから，ここではアキマの方法というものを用いることにする。

アキマの方法もSPLINEの異常屈曲点をなくすために考えられたもので，与えられた2点と幾何学的に求められた1次の微分

微分とは

中学生以下の人には馴染みがない概念でしょうが，ここでは曲線（関数のグラフなど）の傾きを求めることと思っておいてください。といっても曲線は「曲がって」いますから厳密には傾きはありません。そこで，曲線をごく小さな区間で区切った場合には直線に近似できるとして計算することを微分するといいます。もちろん，これは関数が滑らかに連続している場合にのみ有効で，双曲線の原点のように非連続な点では微分不可能です。

関数$y=x^2$では任意の点（x, x^2）で接する直線の傾きy'は常に$y'=2x$で表されます。同様に$y=x^3$では$y'=3x^2$，$y=x^4$では$y'=4x^3$となります。当然，$y=x^{100}$では$y'=100x^{99}$です。n次の関数に対してははっきりした規則性がありますね（証明はしません）。

$$y=111x^5+7x^3+12.5x+10$$

といった，少しこみいった関数も，

$$y'=555x^4+21x^2+12.5$$

のようにそのまま計算可能です。最後の10の部分は接片にあたる部分ですから，傾きを求める際には無視されます。

$y=\log x$とか$y=\sin x$といったかたちの関数も$y'=1/x$，$y'=-\cos x$というぐあいに微分できますが，ここではなぜこうなるのかといった疑問に答えることは割愛しましょう。とにかく，そういう考え方で式が展開されていると思っておいてください。

リスト3　波形を入力

```
1000 int t,r=256,e,w,x2,y2,n(19999),hz=83
1010 float z,q,m(99),tt(99),xx(99),f(99),a0(99),a1(99),a2(99),a3(99)
1020 char a(7800)
1030 float p
1040 str k
1050 screen 2,0,1,1
1060 console ,,0
1070 mouse(1)
1080 mouse(4)
1090 msarea(193,128,576,383)
1100 box(192,128,576,383,15)
1110 line(192,256,576,256,14)
1120 line(384,128,384,383,14)
1130 setmspos(193,256)
1140 int x,y,x1=192,y1=256,ls,rs
1150 xx(0)=192
1160 f(0)=256
1170 for t=1 to 32
1180   print t
1190   msarea(191+t*12,128,192+t*12,383)
1200   repeat
1210     msstat(x,y,ls,rs)
1220     mspos(x,y)
1230     if ls=-1 then{
1240       if t=32 then y=256
1250       pset(x,y,13)
1260       xx(t)=x
1270       f(t)=y
1280       for r=0 to 199
1290       next
1300     }
1310   until ls=-1
1320 next
1330 for t=0 to 32
1340   m(t)=(f(c(t+1))-f(t))/(xx(c(t+1))-xx(t))
1350 next
1360 for t=0 to 32
1370   r=abs(m(c(t+1))-m(t))*m(c(t-1))+abs(m(c(t-1))-m(c(t-2)))*m(t)
1380   e=(abs(m(c(t+1))-m(t))+abs(m(c(t-1))-m(c(t-2))))
1390   if e=0 then tt(t)=(m(c(t-1))+m(t))/2 else tt(t)=r/e
1400 next
1410 /*
1420 for t=0 to 31
1430   a0(t)=f(t)
1440   a1(t)=tt(t)
1450   a2(t)=(3*(f(c(t+1))-f(t))/(xx(c(t+1))-xx(t))-2*tt(t)-tt(c(t+1)))/(xx(c(t
+1))-xx(t))
1460   a3(t)=(tt(t)+tt(c(t+1))-2*(f(c(t+1))-f(t))/(xx(c(t+1))-xx(t)))/pow(xx(c(
t+1))-xx(t),2)
1470   for r=0 to 11
1480     pset(192+t*12+r,a0(t)+a1(t)*r+a2(t)*r*r+a3(t)*r*r*r,11)
1490   next
1500 next
1510 q=15600#/hz
1520 p=q/32:r=0
1530 for t=0 to 15599
1540   e=(t/p) mod 32
1550   z=t/p-int(t/p/32)*32
1560   z=(z-int(z))*12
1570   n(t)=a0(e)+a1(e)*z+a2(e)*z*z+a3(e)*z*z*z
1580 next
1590 for t=0 to 7800
1600   e=n(t*2+1)-n(t*2)
1610   if e<0 then e=8-e:if e>15 then e=15
1620   r=n(t*2+2)-n(t*2+1)
1630   if r<0 then r=8-r:if r>15 then r=15
1640   a(t)=e*16+r
1650 next
1660 a_play(a,4,3)
1670 print"a_play(a,4,3)
1680 end
1690 func c(t)
1700   if t=32 then return(t)
1710   t=(t+32) mod 32
1720   return(t)
1730 endfunc
```

値により，3次の多項式を決定するものである。この方法では与えられた分点を通る手書きの曲線に近い自然なものを得ることができる。加えて，ほかの補間法よりもプログラム化が簡単であるという特徴を持っている。

それではアキマの方法のアルゴリズムを解説してみよう。ここから先は微分法の知識がないとわかりにくいかもしれない。

アキマの方法

いま与えられた点を$(x_0,\ f_0)$，$(x_1,\ f_1)$，……$(x_n,\ f_n)$として，分割された小区間を$[x_j,\ x_{j+1}]$（ただし，$0\leqq j\leqq n-1$）とする。このときこの閉区間の区分多項式P_jを，

$$P_j=a_{0j}+a_{1j}(x-x_j)+a_{2j}(x-x_j)^2+a_{3j}(x-x_j)^3$$

とするとき，a_{0j}，a_{1j}，a_{2j}，a_{3j}は，

$$P_j(x_j)=f_j$$
$$P_j'(x_j)=t_j$$
$$P_j(x_{j+1})=f_{j+1}$$
$$P_j'(x_{j+1})=t_{j+1}$$

により決定される。t_jとは点x_jにおける勾配であり，アキマの方法の特徴的な変数といえる。それではアキマの方法により，上の区分多項式を解くために必要な各係数を求めてみよう。

まず，x_jの両側に分点を2個ずつ加えてこれを図3のようにA, B, C, D, Eとする。これらの点による線分の勾配をm_{j-2}，m_{j-1}，m_j，m_{j+1}とすると，

$$m_i=\frac{f_{i+1}-f_i}{x_{i+1}-x_i}$$

となる（ただし，$j-2\leqq i\leqq j+1$とする）。

これにより，x_jにおける勾配t_jをx_jの両側の線分の比例配分と考えて，

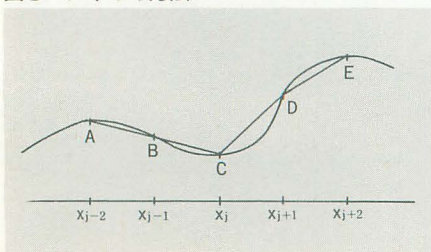
$$t_j=\frac{|m_{j+1}-m_j|m_{j-1}+|m_{j-1}+m_{j-2}|m_j}{|m_{j+1}-m_j|+|m_{j-1}-m_{j-2}|}$$

とする。ただし，式の分母が0になるときには，

$$t_j=\frac{m_{j+1}+m_j}{2}$$

として処理する。

さて，先ほどの条件式を検討すると，

図3　アキマの方法

$$P_j(x_j)=f_j$$

より，

$$a_{0j}+a_{1j}(x_j-x_j)+a_{2j}(x_j-x_j)^2+a_{3j}(x_j-x_j)^3=f_j$$
$$\therefore a_{0j}=f_j$$

また，

$$P_j'(x)=a_{1j}+2a_{2j}(x-x_j)+3a_{3j}(x-x_j)^2$$

よって，$P_j'(x_j)=t_j$は，

$$a_{1j}+2a_{2j}(x_j-x)+3a(x_j-x_j)=t_j$$
$$\therefore a_{1j}=t_j$$

となり，$P_j(x_{j+1})=f_{j+1}$は，

$$a_{0j}+a_{1j}(x_{j+1}-x_j)+a_{2j}(x_{j+1}-x_j)^2+a_{3j}(x_{j+1}-x_j)^3=f_{j+1}$$

$a_{0j}=f_j$，$a_{1j}=t_j$を代入して，

$$f_j+t_j(x_{j+1}-x_j)+a_{2j}(x_{j+1}-x_j)^2+a_{3j}(x_{j+1}-x_j)^3=f_{j+1} \quad \cdots\cdots(1)$$

となる。

$P_j'(x_{j+1})=t_{j+1}$は，

$$a_{1j}-2a_{2j}(x_{j+1}-x_j)+3a_{3j}(x_{j+1}-x_j)^2=t_{j+1}$$

となり，$a_{1j}=t_j$を代入して，

$$t_j+2a_{2j}(x_{j+1}-x_j)+3a_{3j}(x_{j+1}-x_j)^2=t_{j+1} \quad \cdots\cdots(2)$$

となる。

(1)，(2)の式において，$a_{2j}=X$，$a_{3j}=Y$，$x_{j+1}-x_j=A$と置いて連立すると，

$$f_j+t_jA+XA^2+YA^3=f_{j+1} \quad \cdots\cdots(1)'$$
$$t_j+2XA+3YA^2=t_{j+1} \quad \cdots\cdots(2)'$$

となる。

ここで，$3\times(1)'-A\times(2)'$は，

$$\begin{array}{rl} & 3f_j+3t_jA+3XA^2+3YA^3=3f_{j+1} \\ -) & t_jA+2XA^2+3YA^3=t_{j+1}A \\ \hline & 3f_j+2t_jA+XA^2=3f_{j+1}-t_{j+1}A \end{array}$$

$$XA^2=3f_{j+1}-3f_j-2t_jA-t_{j+1}A$$

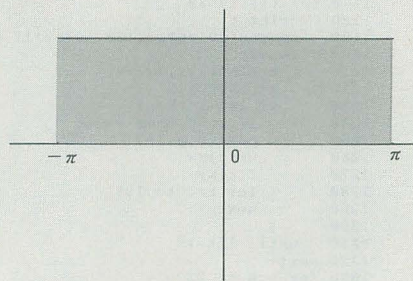
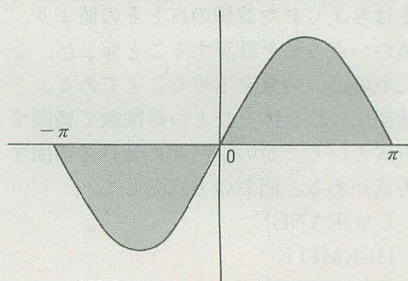
$$X=\frac{3(f_{j+1}-f_j)-A(2t_j+t_{j+1})}{A^2}$$

また，$2\times(1)'-A\times(2)'$は，

$$\begin{array}{rl} & 2f_j+2t_jA+2XA^2+2YA^2=2f_{j+1} \\ -) & t_jA+2XA^2+3YA^3=t_{j+1}A \\ \hline & 2f_j+t_jA-YA^3=2f_{j+1}-t_{j+1}A \end{array}$$

$$YA^3=2f_j+t_jA-2f_{j+1}+t_{j+1}A$$

図4　基本的な関数の積分

(1) $\int_{-\pi}^{\pi}1\,dx=2\pi$

(2) $\int_{-\pi}^{\pi}\cos x\,dx=0$

(3) $\int_{-\pi}^{\pi}\sin x\,dx=0$

(4) $\int_{-\pi}^{\pi}\cos x\cdot\cos x\,dx=\pi$

(5) $\int_{-\pi}^{\pi}\sin x\cdot\sin x\,dx=\pi$

(6) $\int_{-\pi}^{\pi}\cos x\cdot\sin x\,dx=0$

$$Y=\frac{A(t_j+t_{j+1})+2(f_j+f_{j+1})}{A^3}$$

よって，

$$a_{2j}=\frac{3(f_{j+1}-f_j)-(x_{j+1}-x_j)(2t_j+t_{j+1})}{(x_{j+1}-x_j)^2}$$

$$=\frac{3\dfrac{f_{j+1}-f_j}{x_{j+1}-x_j}-2t_j-t_{j+1}}{x_{j+1}-x_j}$$

$$a_{3j}=\frac{(x_{j+1}-x_j)(t_j+t_{j+1})+2(f_j-f_{j+1})}{(x_{j+1}-x_j)^3}$$

$$=\frac{t_j+t_{j+1}-2\dfrac{f_{j+1}-f_j}{x_{j+1}-x_j}}{(x_{j+1}-x_j)^2}$$

となり，これですべての係数が決定した。

アキマの方法では区間 $[x_j,\ x_{j+1}]$ の条件を求めるためには，区間外の点が両側に2つずつ必要である。したがって，端点では勾配が決められないので，点を追加する必要がある。しかし，ここで扱うのは音声データであり，周期関数といえるのでそういった処理は省略することができる。

これで，すべての小区間において区分多項式 $P_j(x)$ を求めることができるようになり，連続的な補間が可能になった。理解していただけただろうか？

波形解析

さて，すでに音が疑似的な周期関数であること，ある条件のもとでは周期関数が三角関係の和として表されることを述べた。これを利用して，先ほどサンプリングし関数のかたちに変換された音声を波形成分に分解してみよう。これにはフーリエ級数展開と呼ばれる手法を用いる。実際，任意の連続関数は周期無限大の周期関数とみなしてフーリエ級数に展開することが可能である。

m，nを自然数とするとき，

$$\int_{-\pi}^{\pi}1dx=2\pi \qquad (1)$$

$$\int_{-\pi}^{\pi}\cos nx\ dx=0,\int_{-\pi}^{\pi}\sin mx\ dx=0 \qquad (2,3)$$

$$\int_{-\pi}^{\pi}\cos mx\cdot\cos nx\ dx=\pi\ (m=n)$$
$$〃\qquad =0\ (m\neq n) \qquad (4)$$

$$\int_{-\pi}^{\pi}\sin mx\cdot\sin nx\ dx=\pi\ \ (m=n)$$
$$〃\qquad =0\ \ (m\neq n) \qquad (5)$$

$$\int_{-\pi}^{\pi}\cos mx\cdot\sin nx\ dx=0 \qquad (6)$$

式の頭にある曲がった記号はインテグラルといい，積分を表す。積分について習っていない人はとりあえず，グラフの面積を

図5　フーリエ級数展開の実際

$$\frac{a_0}{2}=\sum_{j=0}^{max-1}\left\{\frac{1}{T}\int_{x_j}^{x_{j+1}}(a_{0j}+a_{1j}(x-x_j)+a_{2j}(x-x_j)^2+a_{3j}(x-x_j)^3)dx\right\}$$

$$=\sum_{j=0}^{max-1}\left\{\frac{1}{T}\left[a_{0j}x+\frac{1}{2}a_{1j}((x-x_j)^2-x_j^2)+\frac{1}{3}a_{2j}((x-x_j)^3+x_j^3)+\frac{1}{4}a_{3j}((x-x_j)^4-x_j^4)\right]_{x_j}^{x_{j+1}}\right\}$$

$$=\sum_{j=0}^{max-1}\left\{\frac{1}{T}\left((a_{0j}x_{j+1}+\frac{1}{2}a_{1j}((x_{j+1}-x_j)^2-x_j)+\frac{1}{3}a_{2j}((x_{j+1}-x_j)^3+x_j^3)+\frac{1}{4}a_{3j}((x_{j+1}-x_j)^4-x_j^4))\right.\right.$$
$$\left.\left.-(a_{0j}x_j-\frac{1}{2}a_{1j}x_j^2+\frac{1}{3}a_{2j}x_j^3-\frac{1}{4}a_{3j}x_j^4)\right)\right\}$$

$$=\sum_{j=0}^{max-1}\left\{\frac{1}{T}\left(a_{0j}(x_{j+1}-x_j)+\frac{1}{2}a_{1j}(x_{j+1}-x_j)^2+\frac{1}{3}a_{2j}(x_{j+1}-x_j)^3+\frac{1}{4}a_{3j}(x_{j+1}-x_j)^4\right)\right\}$$

$$a_n=\sum_{j=0}^{max-1}\left\{\frac{2}{T}\int_{x_j}^{x_{j+1}}(a_{0j}+a_{1j}(x+x_j)+a_{2j}(x-x_j)^2+a_{3j}(x-x_j)^3)\cos\frac{2n\pi}{T}x\ dx\right\}$$

$$=\sum_{j=0}^{max-1}\left\{\frac{1}{n\pi}\left[(a_{0j}+a_{1j}(x_{j+1}-x_j)+a_{2j}(x-x_j)^2+a_{3j}(x-x_j)^3)\sin\frac{2n\pi}{T}x\right]_{x_j}^{x_{j+1}}-\frac{1}{n\pi}\int_{x_j}^{x_{j+1}}(a_{1j}+2a_{2j}(x-x_j)+3a_{3j}(x-x_j)^2)\sin\frac{2n\pi}{T}x\ dx\right\}$$

$$=\sum_{j=0}^{max-1}\left\{\frac{1}{n\pi}\left(\left((a_{0j}+a_{1j}(x_{j+1}-x_j)+a_{2j}(x_{j+1}-x_j)^2+a_{3j}(x_{j+1}-x_j)^3)\sin\frac{2n\pi}{T}x_{j+1}\right)-\left(a_{0j}\sin\frac{2n\pi}{T}x_j\right)\right)+\frac{T}{2n^2\pi^2}\left[(a_{1j}+2a_{2j}(x-x_j)\right.\right.$$
$$\left.\left.+3a_{3j}(x-x_j)^2)\cos\frac{2n\pi}{T}x\right]_{x_j}^{x_{j+1}}-\frac{T}{2n^2\pi^2}\int_{x_j}^{x_{j+1}}(2a_{2j}+6a_{3j}(x-x_j))\cos\frac{2n\pi}{T}x\ dx\right\}$$

$$=\sum_{j=0}^{max-1}\left\{\frac{1}{n\pi}\left(\left((a_{0j}+a_{1j}(x_{j+1}-x_j)+a_{2j}(x_{j+1}-x_j)^2+a_{3j}(x_{j+1}-x_j)^3)\sin\frac{2n\pi}{T}x_{j+1}\right)-\left(a_{0j}\sin\frac{2n\pi}{T}x_j\right)\right)+\frac{T}{2n^2\pi^2}\left((a_{1j}+2a_{2j}(x_{j+1}-x_j)\right.\right.$$
$$\left.\left.+3a_{3j}(x_{j+1}-x_j)^2)\cos\frac{2n\pi}{T}x_{j+1}-a_{1j}\cos\frac{2n\pi}{T}x_j\right)-\frac{T^2}{4n^3\pi^3}\left[(2a_{2j}+6a_{3j}(x-x_j))\sin\frac{2n\pi}{T}x\right]_{x_j}^{x_{j+1}}+\frac{T^2}{4n^3\pi^3}\int_{x_j}^{x_{j+1}}6a_{3j}\sin\frac{2n\pi}{T}x\ dx\right\}$$

$$=\sum_{j=0}^{max-1}\left\{\frac{1}{n\pi}\left(\left((a_{0j}+a_{1j}(x_{j+1}-x_j)+a_{2j}(x_{j+1}-x_j)^2+a_{3j}(x_{j+1}-x_j)^3)\sin\frac{2n\pi}{T}x_{j+1}\right)-a_{0j}\sin\frac{2n\pi}{T}x_j\right)+\frac{T}{2n^2\pi^2}\left((a_{1j}+2a_{2j}(x_{j+1}-x_j)\right.\right.$$
$$+3a_{3j}(x_{j+1}-x_j)^2)\cos\frac{2n\pi}{T}x_{j+1}-a_{1j}\cos\frac{2n\pi}{T}x_j\right)-\frac{T^2}{4n^3\pi^3}\left((2a_{2j}+6a_{3j}(x_{j+1}-x_j))\sin\frac{2n\pi}{T}x_{j+1}-2a_{2j}\sin\frac{2n\pi}{T}x_j\right)$$
$$\left.-\frac{T^3}{8n^4\pi^4}\left(6a_{3j}\cos\frac{2n\pi}{T}x_{j+1}-6a_{3j}\cos\frac{2n\pi}{T}x_j\right)\right\}$$

$$bn=\sum_{j=0}^{max-1}\left\{\frac{2}{T}\int_{x_j}^{x_{j+1}}(a_{0j}+a_{1j}(x-x_j)+a_{2j}(x-x_j)^2+a_{3j}(x-x_j)^3)\sin\frac{2n\pi}{T}x\ dx\right\}$$

$$=\sum_{j=0}^{max-1}\left\{-\frac{1}{n\pi}\left[(a_{0j}+a_{1j}(x-x_j)+a_{2j}(x-x_j)^2+a_{3j}(x-x_j)^3)\cos\frac{2n\pi}{T}x\right]_{x_j}^{x_{j+1}}\right.$$
$$\left.+\frac{1}{n\pi}\int_{x_j}^{x_{j+1}}(a_{1j}+2a_{2j}(x-x_j)+3a_{3j}(x-x_j)^2)\cos\frac{2n\pi}{T}dx\right\}$$

$$=\sum_{j=0}^{max-1}\left\{-\frac{1}{n\pi}\left((a_{0j}+a_{1j}(x_{j+1}-x_j)+a_{2j}(x_{j+1}-x_j)^2+a_{3j}(x_{j+1}-x_j)^3)\cos\frac{2n\pi}{T}x_{j+1}-a_{0j}\cos\frac{2n\pi}{T}x_j\right)\right.$$
$$+\frac{T}{2n^2\pi^2}\left[(a_{1j}+2a_{2j}(x-x_j)+3a_{3j}(x-x_j)^2)\sin\frac{2n\pi}{T}x\right]_{x_j}^{x_{j+1}}$$
$$\left.-\frac{T}{2n^2\pi^2}\int_{x_j}^{x_{j+1}}(2a_{2j}+6a_{3j}(x-x_j))\sin\frac{2n\pi}{T}x\ dx\right\}$$

$$=\sum_{j=0}^{max-1}\left\{-\frac{1}{n\pi}\left((a_{0j}+a_{1j}(x_{j+1}-x_j)+a_{2j}(x_{j+1}-x_j)^2+a_{3j}(x_{j+1}-x_j)^3)\cos\frac{2n\pi}{T}x_{j+1}-a_{0j}\cos\frac{2n\pi}{T}x_j\right)\right.$$
$$+\frac{T}{2n^2\pi^2}\left((a_{1j}+2a_{2j}(x_{j+1}-x_j)+3a_{3j}(x_{j+1}-x_j)^2)\sin\frac{2n\pi}{T}x_{j+1}-a_{1j}\sin\frac{2n\pi}{T}x_j\right)$$
$$\left.+\frac{T^2}{4n^3\pi^3}\left[(2a_{2j}+6a_{3j}(x-x_j))\cos\frac{2n\pi}{T}x\right]_{x_j}^{x_{j+1}}-\frac{T^2}{4n^3\pi^3}\int_{x_j}^{x_{j+1}}6a_{3j}\cos\frac{2n\pi}{T}x\ dx\right\}$$

$$=\sum_{j=0}^{max-1}\left\{-\frac{1}{n\pi}\left((a_{0j}+a_{1j}(x_{j+1}-x_j)+a_{2j}(x_{j+1}-x_j)^2+a_{3j}(x_{j+1}-x_j)^3)\cos\frac{2n\pi}{T}x_{j+1}-a_{0j}\cos\frac{2n\pi}{T}x_j\right)\right.$$
$$+\frac{T}{2n^2\pi^2}\left((a_{1j}+2a_{2j}(x_{j+1}-x_j)+3a_{3j}(x_{j+1}-x_j)^2)\sin\frac{2n\pi}{T}x_{j+1}-a_{1j}\sin\frac{2n\pi}{T}x_j\right)$$
$$\left.+\frac{T^2}{4n^3\pi^3}\left((2a_{2j}+6a_{3j}(x_{j+1}-x_j))\cos\frac{2n\pi}{T}x_{j+1}-2a_{2j}\cos\frac{2n\pi}{T}x_{j+1}\right)-\frac{T^3}{8n^4\pi^4}\left[6a_{3j}\sin\frac{2n\pi}{T}x\right]_{x_j}^{x_{j+1}}\right\}$$

$$=\sum_{j=0}^{max-1}\left\{-\frac{1}{n\pi}\left((a_{0j}+a_{1j}(x_{j+1}-x_j)+a_{2j}(x_{j+1}-x_j)^2+a_{3j}(x_{j+1}-x_j)^3)\cos\frac{2n\pi}{T}x_{j+1}-a_{0j}\cos\frac{2n\pi}{T}x_j\right)\right.$$
$$+\frac{T}{2n^2\pi^2}\left((a_{1j}+2a_{2j}(x_{j+1}-x_j)+3a_{3j}(x_{j+1}-x_j)^2)\sin\frac{2n\pi}{T}x_{j+1}-a_{1j}\sin\frac{2n\pi}{T}x_j\right)$$
$$+\frac{T^2}{4n^3\pi^3}\left((2a_{2j}+6a_{3j}(x_{j+1}-x_j))\cos\frac{2n\pi}{T}x_{j+1}-2a_{2j}\cos\frac{2n\pi}{T}x_j\right)$$
$$-\frac{6a_{3j}T^3}{8n^4\pi^4}\left(\sin\frac{2n\pi}{T}x_{j+1}-\sin\frac{2n\pi}{T}x_j\right)$$

求めることだと思っておいてほしい。後ろのdxというのは変数xについて積分するということを表すもので縁起ものと考えておけばよい。これらの式ではxが$-\pi\sim\pi$までの区間で各関数のx軸と囲む面積がいくらであるかを表している。x軸より下の部分はマイナスの面積として考えてほしい。

というわけで，(1)から(6)までの方程式のグラフを図4に示す。ちなみにm，nは自然数であればなんでもよいので，もっとも単純なかたち，n=1，m=1とした。

いま，関数f(x)がすべての実数値に対して定義され，周期が2πであるとする。このときf(x)が三角級数，

$$f(x)=\frac{a_0}{2}+\sum_{n=1}^{\infty}(a_n\cos nx+b_n\sin nx)$$

$$=\frac{a_0}{2}+(a_1\cos x+b_1\sin x)$$

$$+(a_2\cos 2x+b_2\sin 2x)+\cdots\cdots$$

$$+(a_n\cos nx+b_n\sin nx)+\cdots\cdots$$

というかたちで表されたとする。この式の両辺に1，cos mx，sin mxをそれぞれかけて，形式的に項別積分できるものと考えて計算すると，

$$\int_{-\pi}^{\pi}f(x)dx=\frac{a_0}{2}\int_{-\pi}^{\pi}dx=\pi a_0 \qquad (1)$$

$$\int_{-\pi}^{\pi}f(x)\cos mx\ dx$$

$$=\sum_{n=1}^{\infty}a_n\int_{-\pi}^{\pi}\cos nx\cdot\cos mx\ dx=\pi a_m \qquad (2,3,5)$$

$$\int_{-\pi}^{\pi}f(x)\sin mx\ dx$$

$$=\sum_{n=1}^{\infty}b_n\int_{-\pi}^{\pi}\sin nx\cdot\sin mx\ dx=\pi b_m \qquad (2,4,5)$$

となる。

これらをまとめると周期2πの関数は，

$$f(x)\simeq\frac{a_0}{2}+\sum(a_n\cos nx+b_n\sin nx)$$

ただし，

$$\begin{cases} a_n=\dfrac{1}{\pi}\displaystyle\int_{-\pi}^{\pi}f(x)\cos nx\ dx & (n=0,1,2) \\ b_n=\dfrac{1}{\pi}\displaystyle\int_{-\pi}^{\pi}f(x)\sin nx\ dx & (n=1,2,3) \end{cases}$$

と表される。これがフーリエ級数展開だ。

さて，フーリエ級数の展開法はわかったが，ここで用いるのは区間2πではない関数の展開法である。そこで区間が任意と考えた場合のフーリエ級数展開を考えてみよう。

まず，周期をTと置く。すると基準の区間が［−1/2T，1/2T］となるので，以下のように変数変換を行う。

$$\xi=\frac{2\pi}{T}x,\quad x=\frac{T}{2\pi}\xi$$

この結果，xの区間［−1/2T，1/2T］はξの区間［$-\pi$，π］に移される。さて，このとき，

$$d\xi=\frac{2\pi}{T}dx$$

であることを考えて先ほどの式を書き換えてみると，周期Tなる関数f(x)のフーリエ級数は，

$$f(x)\simeq\frac{a_0}{2}+\sum_{n=0}^{\infty}\left(a_n\cos\frac{2n\pi}{T}x+b_n\sin\frac{2n\pi}{T}x\right)$$

ただし，

$$\begin{cases} a_n=\dfrac{2}{T}\displaystyle\int_{-1/2T}^{1/2T}f(x)\cos\frac{2n\pi}{T}x\ dx & (n=0,1,2,\cdots\cdots) \\ b_n=\dfrac{2}{T}\displaystyle\int_{-1/2T}^{1/2T}f(x)\sin\frac{2n\pi}{T}x\ dx & (n=1,2,3,\cdots\cdots) \end{cases}$$

となる。

これを実際にアキマの方法で算出した関数に適用すれば，求める波形分析が可能になるわけだ。ということで，また計算だが今回の式は複雑なのでとても文中には挿入できない(難解ではないが)。詳しくは図5をじっくりと見ていただきたい。

なお，リスト4はリスト2と同様な使用方法だが，画面上の赤い線がsin成分，青い線がcos成分を表す。データは32倍波までしか取っていない（それ以上が必要とは思われないが)。ものすごく汚いプログラムなので解析は元の数式を頼りに行ってほしい。間違ってもリストを読もうとしてはいけない。

最後に

今回はさまざまな方法によりADPCMで音を合成することを試みた。特に中心となったのは周波数変調についてだが，この部分さえクリアしてしまえば音量調節や経時的変化などは比較的簡単に解決できるものである。

AD PCMの問題点は量子化ビット数が4ビットと小さく，音質がいまいちということであろう。これはサンプリング周波数を上げることができれば解決される問題だ。たとえばサンプリング周波数が現在の8倍であれば，理論的にはCDと同等の音質を得ることができる。つまり，1/8倍速でサンプリングして処理し，8倍速で再生してやればよいわけだ(?)。こういったものは実用的ではないが，やろうと思えばサンプリングに不可能はないのだ。要は心意気。これを機会にX68000のもうひとつの音楽デバイスを見直してみてほしい。

リスト4　波形分析

```
1000 int t,r=256,e,w,x,y,x1,y1,x2,y2,n(19999),hz=83
1010 float z,q,m(99),tt(99),xx(99),f(99),a0(99),a1(99),a2(99),a3(99)
1020 char a(7800)
1030 float p,pai=3.1415926535898#
1040 str k
1050 fread(a,7800,fopen("bach.pcm","r"))
1060 fcloseall()
1070 screen 2,0,1,1
1080 for t=1 to 383
1090   w=r
1100   e=a(t+0) shr 4
1110   if e>7 then e=8-e
1120   r=r+e
1130   line(t*2-1,w,t*2,r,15)
1140   n(t*2)=r
1150   w=r
1160   e=a(t) and 15
1170   if e>7 then e=8-e
1180   r=r+e
1190   line(t*2,w,t*2+1,r,15)
1200   n(t*2+1)=r
1210 next
1220 /*k=inkey$
1230 a_play(a,4,3)
1240 print"start point"
1250 repeat
1260   mspos(x,y)
1270   msstat(w,w,r,w)
1280 until r=-1
```

```
 1290 x1=x
 1300 y1=y
 1310 print x,y
 1320 for t=0 to 500
 1330 next
 1340 print"end point"
 1350 repeat
 1360   mspos(x,y)
 1370   msstat(w,w,r,w)
 1380 until r=-1
 1390 x2=x
 1400 y2=y
 1410 print x,y
 1420 r=x2-x1
 1430 p=1#*(y2-y1)/r
 1440 for t=x1 to x2
 1450   n(t)=n(t)-p*(t-x1)
 1460   pset(t,n(t),13)
 1470 next
 1480 screen 2,0,1,1
 1490 console ,,0
 1500 box(192,128,576,383,15)
 1510 line(192,256,576,256,14)
 1520 line(384,128,384,383,14)
 1530 p=r/32#
 1540 f(0)=256
 1550 for t=0 to 32
 1560   xx(t)=192+t*12
 1570   f(t)=n(x1+p*t)
 1580 next
 1590 for t=0 to 32
 1600   m(t)=(f(c(t+1))-f(t))/(xx(c(t+1))-xx(t))
 1610 next
 1620 for t=0 to 32
 1630   r=abs(m(c(t+1))-m(t))*m(c(t-1))+abs(m(c(t-1))-m(c(t-2)))*m(t)
 1640   e=(abs(m(c(t+1))-m(t))+abs(m(c(t-1))-m(c(t-2))))
 1650   if e=0 then tt(t)=(m(c(t-1))+m(t))/2 else tt(t)=r/e
 1660 next
 1670 /*
 1680 for t=0 to 31
 1690   a0(t)=f(t)
 1700   a1(t)=tt(t)
 1710   a2(t)=(3*(f(c(t+1))-f(t))/(xx(c(t+1))-xx(t))-2*tt(t)-tt(c(t+1)))/(xx(c(t
+1))-xx(t))
 1720   a3(t)=(tt(t)+tt(c(t+1))-2*(f(c(t+1))-f(t))/(xx(c(t+1))-xx(t)))/pow(xx(c(
t+1))-xx(t),2)
 1730   for r=0 to 11
 1740     pset(192+t*12+r,a0(t)+a1(t)*r+a2(t)*r*r+a3(t)*r*r*r,11)
 1750   next
 1760 next
 1770 /*
 1780 wipe()
 1790 cls
 1800 print"解析の結果・・・";
 1810 z=(x2-x1)/15600#
 1820 box(0,0,767,16,15)
 1830 line(0,256,767,256,15)
 1840 print"周波数：";1/z
 1850 p=0
 1860 for t=0 to 32
 1870   xx(t)=z/32*t
 1880 next
 1890 p=0
 1900 q=0
 1910 for t=0 to 32
 1920   tt(t)=0
 1930   m(t)=0
 1940 next
 1950 for t=1 to 32
 1960   for e=0 to 31
 1970     m(t)=m(t)+((a0(e)+a1(e)*(xx(e+1)-xx(e))+a2(e)*pow(xx(e+1)-xx(e),2)+a3(
e)*pow(xx(e+1)-xx(e),3))*sin(2*t*pai*xx(e+1)/z)-a0(e)*sin(2*pai*t*xx(e)/z))/(t*p
ai)
 1980     m(t)=m(t)+z*((a1(e)+2*a2(e)*(xx(e+1)-xx(e))+3*a3(e)*pow(xx(e+1)-xx(e),
2))*cos(2*t*pai*xx(e+1)/z)-a1(e)*cos(2*t*pai*xx(e)/z))/(2*pow(t*pai,2))
 1990     m(t)=m(t)-z*z*((2*a2(e)+6*a3(e)*(xx(e+1)-xx(e)))*sin(2*t*pai*xx(e+1)/z
)-2*a2(e)*sin(2*t*pai*xx(e)/z))/(4*pow(t*pai,3))
 2000     m(t)=m(t)-z*z*z*6*a3(e)*(cos(2*t*pai*xx(e+1)/z)-cos(2*t*pai*xx(e)/z))/
(8*pow(t*pai,4))
 2010     tt(t)=tt(t)-((a0(e)+a1(e)*(xx(e+1)-xx(e))+a2(e)*pow(xx(e+1)-xx(e),2)+a
3(e)*pow(xx(e+1)-xx(e),3))*cos(2*t*pai*xx(e+1)/z)-a0(e)*cos(2*t*pai*xx(e)/z))/(t
*pai)
 2020     tt(t)=tt(t)+z*((a1(e)+2*a2(e)*(xx(e+1)-xx(e))+a3(e)*pow(xx(e+1)-xx(e),
2))*sin(2*t*pai*xx(e+1)/z)-a1(e)*sin(2*t*pai*xx(e)/z))/(2*pow(t*pai,2))
 2030     tt(t)=tt(t)+z*z*((2*a2(e)+6*a3(e)*(xx(e+1)-xx(e)))*cos(2*t*pai*xx(e+1)
/z)-2*a2(e)*cos(2*t*pai*xx(e)/z))/(4*pow(t*pai,3))
 2040     tt(t)=tt(t)-z*z*z*6*a3(e)*(sin(2*t*pai*xx(e+1)/z)-sin(2*t*pai*xx(e)/z)
)/(8*pow(t*pai,4))
 2050   next
 2060 next
 2070 for t=1 to 32
 2080   if p < abs(m(t)) then p=abs(m(t))
 2090   if p < abs(tt(t)) then p=abs(tt(t))
 2100 next
 2110 q=200#/p
 2120 for t=1 to 32
 2130   fill((t-1)*24+6,256,(t-1)*24+11,256-q*m(t),2)
 2140   fill((t-1)*24+12,256,(t-1)*24+17,256-q*tt(t),4)
 2150 next
 2160 end
 2170 func c(t)
 2180   if t=32 then return(t)
 2190   t=(t+32) mod 32
 2200   return(t)
 2210 endfunc
```

数値演算プロセッサの活用 FLOAT3+.X

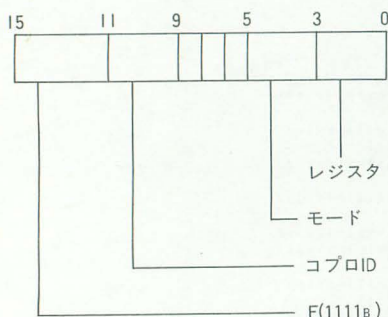
X68000で数値演算といえば，68881を使った数値演算ボードを忘れることはできません。しかし，FLO AT3. Xを導入したものの，いまひとつ実行速度で不満のある方もいるのではないでしょうか。そこで完全コンパチの高速版，FLOAT3+. Xをお届けします。

Nagai Kiyoshi 長井　清
Hirano Teruhiko 平野　照比古
Nakano Shuichi 中野　修一

X68000による実数演算

すでにご存じのように，X68000では浮動小数点演算はFLOATn.Xというデバイスドライバによって実行します。これは68020のコードのうちのFE系列にあたるユーザー定義コプロセッサ命令を用いることでデバイスドライバを呼び出し，演算を実行しています。

もともと，こういった命令は68000にとっては未実装命令ですので実行しようとするとトラップが発生し，所定の例外処理ルーチンに制御が移されます。その飛び先で実数演算をまとめて行えば，デバイスドライバの交換で，浮動小数点演算プロセッサをつけたマシンとつけていないマシンとで，まったく同じオブジェクトコードが実行できるわけです。これがX68000の数値演算の方法です。

ちなみに，68020 のF系列の命令コード(命令コードが16進数のFで始まるもの)は，

のような構成になっており，「FE系列の命令」とはコプロセッサID番号が7番にあたるコプロセッサを動かすための命令に対応します。同様な方法によるOSのファンクションコールはFF系列のコードを使っていましたが，これもID番号7にあたるコプロセッサ用命令エリアを使用しているわけです。Macintoshの Toolbox呼び出しを見てもわかるように，こういった手法は68000系プロセッサのファンクションコールの常套手段といえるものでしょう。

コプロセッサインタフェイス

それでは，こういうとき決まって引き合いに出される，68020のコプロセッサインタフェイスというものについて，ざっと解説してみましょう。

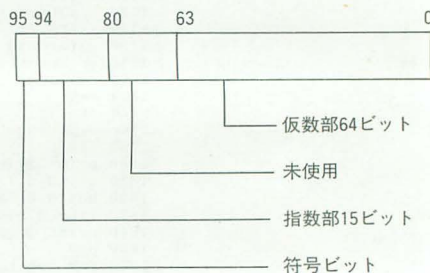
現在，モトローラからコプロセッサとして発表されているのはメモリ管理ユニット68851 (ID0) と浮動小数点演算プロセッサ68881, 68882 (ID1) だけですが，68020ではユーザーが自由にコプロセッサを設計できるように，コプロセッサインタフェイスが公開されています。これはメインプロセッサとどのようにデータをやりとりするか，コプロセッサに必要なハードウェアはなにかなどといった情報をまとめて表したものです。

まず，すべてのコプロセッサは図1のようなレジスタを備えていなければなりません (一部は省略可)。コプロセッサを組み込んである場合，まずMPUは命令をデコードし，F系列の命令かどうかを確認してF系列の命令ならば，コプロセッサのコマンドレジスタにコマンドワードを書き込みます。コプロセッサはコマンドを評価し，パラメータに応じてメインプロセッサと双方向通信を始めます。先ほど示したF系列の命令の直後にはコマンドワードが，その後ろには拡張ワード群が並んでいますが，これらの命令は次々とコプロセッサに送られるわけです。

68020 ではアセンブラの1命令でこれらの動作が自動的に行われますが，68000などではこういった操作をプログラムによってエミュレートしなければなりません。

68881の概要

X68000 の数値演算ボードに取り付けられている 68881 浮動小数点演算プロセッサの演算フォーマットはIEEE754という 2進浮動小数点演算規格に完全準拠しています。また，68881は独自の内部レジスタとして浮動小数点演算用に8つの80ビットレジスタ (FP0～7) とステータスレジスタ，制御レジスタ，命令アドレスレジスタを備えています。

まずは浮動小数点レジスタから見てみましょう。通常，X68000の浮動小数点演算といえば，64ビットの倍精度演算を指しますが，68881の内部ではすべての演算は96ビットの拡張精度演算で行われます (実際には96ビットのうちの80ビット分だけが使われる)。拡張精度のフォーマットを以下に示します。

未使用の部分は将来的な拡張用ということであり，使用する際には0で埋めるように指導されています。

ステータスレジスタや制御レジスタは32ビットのレジスタを1バイトずつ4つに分けて別々の目的に使用されます。これらは例外処理などの際に使用されたり，フラグの役目をするレジスタです。最後の命令アドレスレジスタは 68020 と並行動作している際の浮動少数点例外発生アドレスを調べるために使われます。このあたりの詳しいところは『MC68881ユーザーズ・マニュアル』を参照してください。

命令セットについて

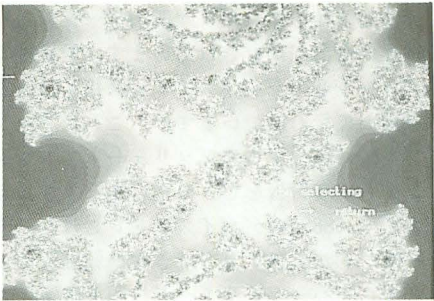
68881は1個のMPUなのですが，数値演算専用に 68020 と組み合わせて使うことを意識して作られているため，ふつうのマイクロプロセッサのようなプログラムを作ることはできません。データの移動や演算，分岐命令はあっても 68881 自体ではアドレス計算は行うことはできず，68020に対して

オペランドを要求するわけです。命令は大きく，入出力（データ転送）命令，レジスタ間データ転送命令，単項/二項演算命令，分岐命令などを備えており，ふつうのCPUが整数の加算を行うように，やすやすと倍精度関数演算をこなしてしまいます。

はっきりいってこの68881という石は利口な石です。内蔵された関数はBASICのそれよりも多く，Hyperbolic tangent（双曲線正接）を求めるといった操作なら，BASICやC言語を使うよりもアセンブラのほうが簡単かもしれません。

FLOAT3+.X

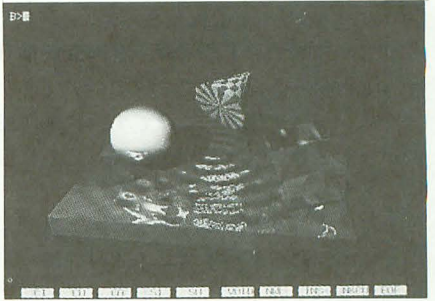
以前，X68000にX1turboのWORD POWERを移植した長井さんによるプログラムです。プログラムの名前からだいたいの見当はつくでしょうが，FLOAT3.Xを解析して無駄と思われる部分を改善したものがこのFLOAT3＋.Xです。基本的なアルゴリズムは同じでも，平均25%程度速度が上がり，かつプログラムサイズも小さくなっています（注：＋は全角です）。

残念ながらソースリストを掲載するスペースがないのですが，62Kバイトに及ぶ逆アセンブルリストと格闘した長井さんにはまったく脱帽です。このプログラムでは明らかに無駄と思われるフラグチェックはしない，できるだけレジスタを使用するように心がけるといった方針でプログラムされています。数値演算プロセッサを入手されている方はぜひ活用してみてください（ダンプリストは6月号のマシン語入力ツールを使用して打ち込んでください）。

自己平方フラクタル

C-TRACE68の実行結果

以下にFLOAT2.X，FLOAT3.X，FLOAT3＋.Xによる演算速度変化を見てみましょう。

より現実的なプログラムで実験してみましょう。Oh！X1988年2月号に掲載された自己平方フラクタルのプログラムを実行した結果，FLOAT2.X（68881なし）では18時間16分4秒，FLOAT3.Xでは9時間20分45秒，そしてFLOAT3＋.Xでは5時間44分31秒となりました。

市販のソフトウェアに対してはどうでしょうか。以下はC-TRACE68でレイトレーシングを行った際の，描画時間を比べたものです(SAMPLE 2, AREA＝512×512, STEP＝2,2)。

FLOAT2.X	9時間26分45秒
FLOAT3.X	3時間18分28秒
FLOAT3＋.X	2時間25分41秒

当然ながら，浮動小数点演算プロセッサなしの場合は圧倒的に遅いことがおわかりでしょう。いまのところ，FLOAT3＋.Xでもほとんどのソフトは支障なく動作するようです。

8086系のマシンではコンパイラなどで8087オプションを指定すると，劇的に高速化されるのに対して，X68000ではそれほど極端には速くなりません（もちろん，80x86マシン用のコプロ対応プログラムをコプロを持たないマシンで実行すれば一生かかっても結果は出てきませんから，一概に遅いとは

図1　コプロセッサのレジスタ

アドレス	名　称	構　成	ビット長	リード／ライト	必須	機　能
$00	レスポンスレジスタ	CA / PC / ファンクション / パラメータ	16	R	○	メインプロセッサへのサービス要求
$02	コントロールレジスタ	不使用 / XA / XB	16	$\overline{W}$	○	エクセプションクリア，アボートの指示
$04	セーブレジスタ	フォーマットコード / バイト長	16	R	○	内部状態退避の開始を指示
$06	リストアレジスタ	フォーマットコード / バイト長	16	R/$\overline{W}$	○	内部状態の回復の開始を指示
$08	オペレーションレジスタ	オペレーションワード	16	$\overline{W}$	○	オペレーションワードを受け取る
$0A	コマンドレジスタ	コマンドワード	16	$\overline{W}$	○	コマンドワードを受け命令を開始
$0E	コンディションレジスタ	不使用 / 条件選択コード	16	$\overline{W}$	○	条件選択コードを受け命令を開始
$10	オペランドレジスタ	各種	32	R/$\overline{W}$	○	あらゆるオペランド，データの受け渡し
$14	レジスタセレクトレジスタ	レジスタ選択コード	16	R		レジスタ選択コードを示す
$18	インストラクションアドレスレジスタ		32	R/$\overline{W}$		コプロセッサ命令のアドレスを格納
$1C	オペランドアドレスレジスタ		32	R/$\overline{W}$		オペランドの実効アドレスを格納

いえませんが)。その代わり，どのマシンでも同じオブジェクトが実行でき，数値演算ボードを組み込むだけですべてのアプリケーションが高速化されるのです。将来，68882ボードと専用ドライバが出現してくれれば，もっと楽しくなることでしょう。

また，今回のFLOAT3+.Xの例を見てもわかるように，デバイスドライバの部分が最適化されていくと，ハードウェアが同じでもすべてのものが高速化されていくのです。こうなると，FLOAT2.Xも最適化すればもっと速くなるのではないかという気がしてきませんか？　というわけで，FLOAT2+.Xなど作ってみたいという方も歓迎いたします。

参考までに68000で68881アセンブリ言語から直接扱うためのマクロ群を紹介しましょう（リスト2 FPPMAC. H)。これは平野さんの投稿によるもので，演算プロセッサに対し直接的な操作をすることで68881の能力を十分に引き出すことができます。

これらのマクロは，具体的にはコプロセッサとメインプロセッサ間のデータ転送を行う命令群が用意されています。数値演算プロセッサのデバイスアドレス指定はA5を用いて行われていますが，これはアセンブル時に発生するアブソリュートアドレシングという警告をできるだけ少なくするためです。68881のデバイスアドレスはユーザーモードからアクセスできない領域なので，必ずスーパーバイザモードに切り換えてから実行せねばなりません。

マクロの定義部を見ればおおよその使い方はわかると思いますが，具体的な使用例として2月号で掲載された自己平方フラクタルのプログラムをリスト1,2に挙げておきます。FLOAT3+.Xが多少速いといっても，石の性能を考えると十倍の速度でもおかしくはありません。参考までにこのマクロを使って68881を直接ドライブしてやる場合，このフラクタル図形を描き終えるまで1時間とかかりません。気持ちがいいほどの高速さです。

実行方法は前もってFRFPP.Sをアセンブルしておき，

CC FRACT.BAS FRFPP.O

として，コンパイラでリンクしてください。これで実行ファイルFRACT.Xができあがります。

このようにすると確かに計算速度は向上するのですが，プログラムの実行には68881が必須となります。どのようなマシンでもまったく同じプログラムが動作するFLOATn. Xの発想に比べると美しくないことは否定できません。自分ひとりで使うことしか考えないのであればともかく，少しでも他人に見せることを考えるなら，ハードウェアチェックを行って2種類の演算処理部を切り換えるといった処理が必要となるでしょう。

参考文献

モトローラ，「MC68881ユーザーズ・マニュアル」，電波新聞社

朝日広治他，「図解32ビットマイクロプロセッサMC68020」，オーム社

W. クレイマー/G.ケイン，「68000ファミリハンドブック」，啓学出版

リスト1 FLOAT3+.X

```
0000  48 55 00 00 00 00 00 00  : 9D
0008  00 00 01 82 00 00 26 68  : 11
0010  00 00 00 00 00 00 00 00  : 00
0018  00 00 02 3C 00 00 00 00  : 3E
0020  00 00 00 00 00 00 00 00  : 00
0028  00 00 00 00 00 00 00 00  : 00
0030  00 00 00 00 00 00 00 00  : 00
0038  00 00 00 00 00 00 00 00  : 00
0040  FF FF FF FF 80 00 00 00  : 7C
0048  00 1A 00 00 00 22 46 4C  : CE
0050  4F 41 54 2A 2F 2D 00 00  : 6A
0058  00 00 23 CD 00 00 00 16  : 06
0060  4E 75 48 E7 80 04 2A 79  : 19
0068  00 00 00 16 42 80 10 2D  : 15
0070  00 02 61 10 1B 40 00 03  : D1
0078  E0 48 1B 40 00 04 4C DF  : B2
-------------------------------------
SUM:  C4 6E 3D 01 8C 17 F2 52  1068

0080  20 01 4E 75 66 1A 48 E7  : 93
0088  40 40 61 00 00 A2 4C DF  : AE
0090  02 02 67 0C 2B 7C 00 00  : 1E
0098  26 68 00 0E 42 80 4E 75  : 21
00A0  30 3C 50 03 4E 75 46 45  : 0D
00A8  66 6E 48 E7 02 06 4D EF  : 47
00B0  00 0E 2A 56 0C 1D 00 FE  : B5
00B8  67 10 4C DF 60 40 2F 39  : AA
00C0  00 00 00 86 4E 75 00 00  : 49
00C8  00 00 42 86 1C 1D 2C CD  : FA
00D0  08 2F 00 05 00 0C 66 02  : B0
00D8  4E 6E E5 46 DC BC 00 00  : 7F
00E0  01 CA 2A 46 2A 55 4E 95  : 9D
00E8  40 C6 1F 46 00 0D 4C DF  : A3
00F0  60 40 08 17 00 07 66 02  : 2E
00F8  4E 73 00 7C 80 00 4E 73  : 7E
-------------------------------------
SUM:  CA 53 9C 24 7F 53 84 5E  B96B

0100  B0 BC 00 00 FE 00 65 22  : F1
0108  B0 BC 00 00 FF 00 64 1A  : E9
0110  48 E7 00 40 90 7C FE 00  : 79
0118  E5 80 43 F9 00 00 01 CA  : 6C
0120  D3 C0 20 11 22 88 22 5F  : EF
0128  4E 75 70 FF 4E 75 72 0B  : 72
0130  43 F9 00 00 00 6A 70 80  : 96
0138  4E 4F 23 C0 00 00 00 86  : 06
0140  22 40 59 89 70 84 4E 4F  : D5
0148  B0 B9 00 00 00 66 66 1A  : 4F
0150  22 79 00 00 00 86 72 0B  : 9E
0158  70 80 4E 4F 48 79 00 00  : 4E
0160  01 94 FF 09 58 8F 42 80  : 46
0168  4E 75 48 79 00 00 08 38  : C4
0170  FF 09 58 8F 70 FF 43 F9  : 9A
0178  00 00 01 4E 22 3C 00 00  : AD
-------------------------------------
SUM:  F1 60 3D 40 9F 96 7F 9B  1C34

0180  01 FF 70 80 4E 4F 23 C0  : 70
0188  00 00 01 7E 4E 75 42 90  : 14
0190  20 10 48 40 33 C0 00 E9  : 94
0198  E0 06 32 00 02 41 FF 00  : 5A
01A0  67 14 41 F9 00 E9 E0 10  : 8E
01A8  12 00 E4 09 48 81 53 41  : 5C
01B0  20 90 51 C9 FF FC 20 79  : 5E
01B8  00 00 01 7E 4E D0 00 00  : 9D
01C0  00 00 61 00 FF 6A 67 0A  : 3B
01C8  42 67 2F 3C 00 00 34 BA  : 02
01D0  FF 31 FF 00 0D 0A 82 B7  : 7F
01D8  82 C5 82 C9 95 82 93 AE  : EA
01E0  8F AC 90 94 93 5F 89 89  : 63
01E8  8E 5A 83 70 83 62 83 50  : 93
01F0  81 5B 83 57 82 CD 93 6F  : 07
01F8  98 5E 82 B3 82 EA 82 C4  : DD
-------------------------------------
SUM:  93 D5 8B 9A 21 69 88 38  34A9

0200  82 A2 82 DC 82 B7 0D 0A  : D2
0208  00 00 00 00 19 90 00 00  : A9
0210  1A 1E 00 00 1A 64 00 00  : B6
0218  00 84 00 00 19 DC 00 00  : 79
0220  1A 88 00 00 16 1C 00 00  : D4
0228  00 84 00 00 06 F0 00 00  : 7A
0230  06 FE 00 00 00 84 00 00  : 88
0238  00 84 00 00 1B 60 00 00  : FF
0240  1B 66 00 00 1B B4 00 00  : 50
0248  00 84 00 00 0A 9C 00 00  : 2A
0250  09 C6 00 00 0B 6E 00 00  : 48
0258  0C 0A 00 00 0B 24 00 00  : 45
0260  0B E0 00 00 0B 9C 00 00  : 92
0268  0B C4 00 00 09 60 00 00  : 38
0270  00 84 00 00 18 BA 00 00  : 56
0278  19 16 00 00 16 40 00 00  : 85
-------------------------------------
SUM:  1B CA 82 DC 82 4F 0D 0A  229F

0280  16 7C 00 00 17 5A 00 00  : 03
0288  16 EE 00 00 0A 54 00 00  : 62
0290  10 24 00 00 0C 5C 00 00  : 9C
0298  09 2C 00 00 0F CC 00 00  : 10
02A0  0F F6 00 00 0E E4 00 00  : F7
02A8  00 84 00 00 1B 2E 00 00  : CD
02B0  1A D6 00 00 1A BC 00 00  : C6
02B8  21 A4 00 00 21 B0 00 00  : 96
02C0  21 BC 00 00 21 C8 00 00  : C6
02C8  21 F0 00 00 1A D0 00 00  : FB
02D0  18 10 00 00 17 8C 00 00  : CB
02D8  17 E4 00 00 18 52 00 00  : 65
02E0  14 F4 00 00 22 DA 00 00  : 04
02E8  22 E6 00 00 22 F2 00 00  : 1C
02F0  22 FE 00 00 23 16 00 00  : 59
02F8  23 0A 00 00 23 3A 00 00  : 8A
-------------------------------------
SUM:  7B 30 00 00 94 E6 00 00  300E

0300  14 CC 00 00 14 DA 00 00  : CE
0308  15 24 00 00 1B 98 00 00  : EC
0310  00 84 00 00 00 84 00 00  : 08
0318  00 84 00 00 00 84 00 00  : 08
0320  00 84 00 00 00 84 00 00  : 08
0328  00 84 00 00 00 84 00 00  : 08
0330  07 18 00 00 07 4A 00 00  : 70
0338  22 76 00 00 22 8A 00 00  : 44
0340  22 9E 00 00 06 EE 00 00  : B4
0348  06 EE 00 00 1C 3E 00 00  : 4E
0350  1C 28 00 00 1C 50 00 00  : B0
0358  1C 62 00 00 1C 0A 00 00  : A4
0360  1C 14 00 00 1C 1E 00 00  : 6A
0368  00 84 00 00 1C 74 00 00  : 14
0370  1C 8A 00 00 1C D6 00 00  : 98
0378  24 80 00 00 24 8C 00 00  : 54
-------------------------------------
SUM:  0E 46 00 00 2A D0 00 00  815E

0380  24 98 00 00 24 A4 00 00  : 84
0388  24 CA 00 00 1C E6 00 00  : F0
0390  1D 2E 00 00 1C EC 00 00  : 53
0398  1D 4C 00 00 1D 6C 00 00  : F2
03A0  1D CC 00 00 25 A4 00 00  : B2
03A8  25 B0 00 00 25 BC 00 00  : B6
03B0  25 C8 00 00 25 E0 00 00  : F2
03B8  25 D4 00 00 26 02 00 00  : 21
03C0  1D F0 00 00 1D F8 00 00  : 22
03C8  1E 0C 00 00 1B EC 00 00  : 31
03D0  00 84 00 00 00 84 00 00  : 08
03D8  00 84 00 00 00 84 00 00  : 08
03E0  00 84 00 00 00 84 00 00  : 08
03E8  00 84 00 00 00 84 00 00  : 08
03F0  07 A6 00 00 07 D2 00 00  : 86
03F8  25 48 00 00 25 5C 00 00  : EE
-------------------------------------
SUM:  75 EE 00 00 72 46 00 00  1248

0400  25 70 00 00 06 EE 00 00  : 89
0408  06 EE 00 00 00 84 00 00  : 78
0410  00 84 00 00 00 84 00 00  : 08
0418  00 84 00 00 00 84 00 00  : 08
0420  00 84 00 00 00 84 00 00  : 08
0428  00 84 00 00 00 84 00 00  : 08
0430  00 84 00 00 00 84 00 00  : 08
0438  00 84 00 00 00 84 00 00  : 08
0440  00 84 00 00 00 84 00 00  : 08
0448  00 84 00 00 00 84 00 00  : 08
0450  00 84 00 00 00 84 00 00  : 08
0458  00 84 00 00 00 84 00 00  : 08
0460  00 84 00 00 00 84 00 00  : 08
0468  00 84 00 00 00 84 00 00  : 08
0470  00 84 00 00 00 84 00 00  : 08
0478  00 84 00 00 00 84 00 00  : 08
-------------------------------------
SUM:  2B 96 00 00 06 AA 00 00  ED88

0480  00 84 00 00 00 84 00 00  : 08
0488  00 84 00 00 00 84 00 00  : 08
0490  00 84 00 00 00 84 00 00  : 08
0498  00 84 00 00 00 84 00 00  : 08
04A0  00 84 00 00 00 84 00 00  : 08
04A8  00 84 00 00 00 84 00 00  : 08
04B0  00 84 00 00 00 84 00 00  : 08
04B8  00 84 00 00 00 84 00 00  : 08
04C0  00 84 00 00 00 84 00 00  : 08
04C8  00 84 00 00 00 84 00 00  : 08
04D0  00 84 00 00 00 84 00 00  : 08
04D8  00 84 00 00 00 84 00 00  : 08
04E0  00 84 00 00 00 84 00 00  : 08
04E8  00 84 00 00 00 84 00 00  : 08
04F0  00 84 00 00 00 84 00 00  : 08
04F8  00 84 00 00 00 84 00 00  : 08
-------------------------------------
SUM:  00 40 00 00 00 40 00 00  D893

0500  00 84 00 00 00 84 00 00  : 08
0508  00 84 00 00 00 84 00 00  : 08
```

```
0510  00 84 00 00 00 84 00 00  : 08
0518  00 84 00 00 00 84 00 00  : 08
0520  00 84 00 00 00 84 00 00  : 08
0528  00 84 00 00 00 84 00 00  : 08
0530  00 84 00 00 00 84 00 00  : 08
0538  00 84 00 00 00 84 00 00  : 08
0540  00 84 00 00 00 84 00 00  : 08
0548  00 84 00 00 00 84 00 00  : 08
0550  00 84 00 00 00 84 00 00  : 08
0558  00 84 00 00 00 84 00 00  : 08
0560  00 84 00 00 00 84 00 00  : 08
0568  00 84 00 00 00 84 00 00  : 08
0570  00 84 00 00 00 84 00 00  : 08
0578  00 84 00 00 00 84 00 00  : 08
-----------------------------------
SUM:  00 40 00 00 00 40 00 00  D893

0580  00 84 00 00 00 84 00 00  : 08
0588  00 84 00 00 05 EE 00 00  : 77
0590  06 04 00 00 06 1A 00 00  : 2A
0598  06 30 00 00 06 46 00 00  : 82
05A0  06 5C 00 00 1F E6 00 00  : 67
05A8  20 1E 00 00 06 88 00 00  : CC
05B0  06 72 00 00 06 9E 00 00  : 1C
05B8  06 B4 00 00 06 CA 00 00  : 8A
05C0  20 76 00 00 20 82 00 00  : 38
05C8  20 8E 00 00 20 9A 00 00  : 68
05D0  20 CC 00 00 06 DC 00 00  : CE
05D8  23 74 00 00 23 80 00 00  : 3A
05E0  23 8C 00 00 23 98 00 00  : 6A
05E8  23 C2 00 00 05 CA 00 00  : B4
05F0  05 DC 00 00 21 60 00 00  : 62
05F8  24 46 00 00 21 6C 00 00  : F7
-----------------------------------
SUM:  30 90 00 00 15 4E 00 00  0117

0600  24 52 00 00 08 2A 00 00  : A8
0608  00 C0 48 E7 C0 00 4C D6  : D1
0610  00 03 61 00 15 5A 4C DF  : FE
0618  00 03 4E 75 48 E7 80 00  : 75
0620  4C D6 00 01 61 00 16 8E  : 28
0628  4C DF 00 01 4E 75 48 E7  : 1E
0630  C0 00 4C D6 00 03 61 00  : 46
0638  13 98 48 D6 00 03 4C DF  : F7
0640  00 03 4E 75 48 E7 C0 00  : B5
0648  4C D6 00 03 61 00 14 10  : AA
0650  48 D6 00 03 4C DF 00 03  : 4F
0658  4E 75 48 E7 C0 00 4C D6  : D4
0660  00 03 61 00 14 40 48 D6  : D6
0668  00 03 4C DF 00 03 4E 75  : F4
0670  48 E7 C0 00 4C D6 00 03  : 14
0678  61 00 13 A2 48 D6 00 03  : 37
-----------------------------------
SUM:  1A 76 A1 ED 31 9B D9 43  0D2B

0680  4C DF 00 03 4E 75 48 E7  : 20
0688  C0 00 4C D6 00 03 61 00  : 46
0690  14 38 48 D6 00 03 4C DF  : 98
0698  00 03 4E 75 48 E7 C0 00  : B5
06A0  4C D6 00 03 61 00 0F B6  : 4B
06A8  48 D6 00 03 4C DF 00 03  : 4F
06B0  4E 75 48 E7 80 00 4C D6  : 94
06B8  00 01 61 00 10 00 48 D6  : 90
06C0  00 01 4C DF 00 01 4E 75  : F0
06C8  48 E7 80 00 4C D6 00 01  : D2
06D0  61 00 0F AE 48 D6 00 01  : 3D
06D8  4C DF 00 01 4E 75 48 E7  : 1E
06E0  C0 00 4C D6 00 03 61 00  : 46
06E8  10 B2 48 D6 00 03 4C DF  : 0E
06F0  00 03 4E 75 48 E7 C0 00  : B5
06F8  4C D6 00 03 61 00 10 30  : C6
-----------------------------------
SUM:  13 8E 48 C3 5E 50 6B 98  3EAF

0700  48 D6 00 03 4C DF 00 03  : 4F
0708  4E 75 48 E7 F0 00 4C D6  : 04
0710  00 0F 61 00 14 02 4C DF  : B1
0718  00 0F 4E 75 48 E7 C0 00  : C1
0720  4C D6 00 03 61 00 15 A4  : 3F
0728  4C DF 00 03 4E 75 4E 75  : B4
0730  48 E7 20 00 61 00 12 F4  : B6
0738  4C DF 00 04 4E 75 4A 81  : BD
0740  67 10 48 E7 30 00 61 00  : 37
0748  13 98 22 02 4C DF 00 0C  : 06
0750  4E 75 00 3C 00 01 4E 75  : C3
0758  24 00 08 82 00 1F 84 81  : D2
0760  67 20 24 00 C4 BC 7F F0  : 9A
0768  00 00 E1 9A E9 9A 94 BC  : 4E
0770  00 00 03 FF C0 BC 80 0F  : 0D
0778  FF FF 80 BC 3F F0 00 00  : 69
-----------------------------------
SUM:  14 20 11 65 1E B3 DD 03  96F7

0780  4E 75 42 80 42 81 42 82  : 0C
0788  4E 75 48 E7 20 00 D4 BC  : A2
0790  00 00 03 FF 2F 02 24 00  : 57
0798  08 82 00 1F 84 81 67 30  : 45
07A0  24 00 C4 BC 7F F0 00 00  : 13
07A8  E1 9A E9 9A 94 BC 00 00  : 4E
07B0  03 FF D4 9F 67 26 B4 BC  : 72
07B8  00 00 07 FF 64 1E E8 9A  : 0A
07C0  E0 9A C0 BC 80 0F FF FF  : 83
07C8  80 82 4C DF 00 04 4E 75  : F4
07D0  58 8F 42 80 42 81 4C DF  : 97
07D8  00 04 4E 75 00 3C 00 01  : 04
07E0  4C DF 00 04 4E 75 22 00  : 14
07E8  C2 BC 7F FF FF FF 67 1E  : 7F
07F0  C2 BC 7F 80 00 00 E1 99  : F7
07F8  E3 99 92 BC 00 00 00 7F  : 49
-----------------------------------
SUM:  17 A4 41 48 02 38 40 4E  6F10

0800  C0 BC 80 7F FF FF 80 BC  : B5
0808  3F 80 00 00 4E 75 42 80  : 44
0810  4E 75 48 E7 40 00 D2 BC  : C0
0818  00 00 00 7F 2F 01 22 00  : D1
0820  C2 BC 7F FF FF FF 67 2E  : 8F
0828  C2 BC 7F 80 00 00 E1 99  : F7
0830  E3 99 92 BC 00 00 00 7F  : 49
0838  D2 9F 67 24 B2 BC 00 00  : 6A
0840  00 FF 64 1C E2 99 E0 99  : 73
0848  C0 BC 80 7F FF FF 80 81  : 7A
0850  4C DF 00 02 4E 75 58 8F  : D7
0858  42 80 4C DF 00 02 4E 75  : B2
0860  00 3C 00 01 4C DF 00 02  : 6A
0868  4E 75 20 3C 49 45 45 45  : 37
0870  22 3C 46 50 43 50 4E 75  : 4A
0878  0D 0A 95 82 93 AE 8F AC  : AA
-----------------------------------
SUM:  51 72 EA CF 07 61 26 C4  8F64

0880  90 94 93 5F 89 89 8E 5A  : 10
0888  20 43 6F 2D 50 52 4F 43  : 33
0890  45 53 53 4F 52 20 44 52  : 42
0898  49 56 45 52 20 66 6F 72  : 9D
08A0  20 58 36 38 30 30 30 20  : 96
08A8  76 65 72 73 69 6F 6E 20  : 26
08B0  31 2E 31 58 0D 0A 81 69  : E9
08B8  82 68 82 64 82 64 82 64  : 9C
08C0  83 74 83 48 81 5B 83 7D  : 9E
08C8  83 62 83 67 81 6A 0D 0A  : D1
08D0  00 00 61 12 64 0E B0 3C  : D1
08D8  00 41 65 08 B0 3C 00 5B  : F5
08E0  0A 3C 00 01 4E 75 B0 3C  : F6
08E8  00 61 65 08 B0 3C 00 7B  : 35
08F0  0A 3C 00 01 4E 75 B0 3C  : F6
08F8  00 30 65 08 B0 3C 00 3A  : C3
-----------------------------------
SUM:  A1 F3 8B 6F 85 DF D1 B9  9F65

0900  0A 3C 00 01 4E 75 B0 3C  : F6
0908  00 30 65 08 B0 3C 00 32  : BB
0910  0A 3C 00 01 4E 75 B0 3C  : F6
0918  00 30 65 08 B0 3C 00 38  : C1
0920  0A 3C 00 01 4E 75 61 CE  : 39
0928  64 1A B0 3C 00 41 65 14  : 24
0930  B0 3C 00 47 65 0A B0 3C  : 8E
0938  00 61 65 08 B0 3C 00 67  : 21
0940  0A 3C 00 01 4E 75 61 9E  : 09
0948  65 04 90 3C 00 20 4E 75  : 18
0950  61 84 65 04 D0 3C 00 20  : 7A
0958  4E 75 61 F4 B0 3C 00 61  : 65
0960  65 04 D0 3C FF D9 D0 3C  : 59
0968  FF D0 4E 75 74 0E 2F 08  : 4B
0970  61 00 05 B2 20 57 61 08  : F8
0978  20 5F 61 00 06 7E 4E 75  : 27
-----------------------------------
SUM:  35 37 B9 36 C6 27 33 BC  1133

0980  10 18 67 12 B0 3C 00 2E  : BB
0988  66 F6 10 10 61 00 FF 68  : 44
0990  64 04 53 88 61 02 4E 75  : 69
0998  10 E8 00 01 66 FA 4E 75  : 1C
09A0  48 E7 E0 40 4F EF FF F0  : 7C
09A8  22 48 20 4F 61 58 20 4F  : 01
09B0  61 1E 74 20 92 00 6B 08  : 18
09B8  67 06 12 C2 53 01 60 F6  : EB
09C0  20 4F 61 1A 20 49 4F EF  : 91
09C8  00 10 4C DF 02 07 4E 75  : 07
09D0  42 80 4A 10 67 06 52 88  : 63
09D8  52 80 60 F6 4E 75 12 D8  : D5
09E0  66 FC 53 89 4E 75 10 18  : 29
09E8  66 06 4A 11 66 12 4E 75  : 02
09F0  12 19 67 06 B0 01 67 EE  : 9E
09F8  4E 75 44 FC 00 00 4E 75  : C6
-----------------------------------
SUM:  FC 3C EF B7 A8 D3 99 71  5D8D

0A00  44 FC 00 01 4E 75 48 E7  : 33
0A08  E1 60 4E 56 FF F6 22 4F  : 4B
0A10  45 FA 00 56 42 47 4A 80  : E8
0A18  6A 04 44 80 46 47 24 12  : F5
0A20  67 12 42 01 52 01 90 82  : 21
0A28  64 FA D0 82 53 01 12 C1  : D7
0A30  58 8A 60 EA 42 01 22 4F  : E0
0A38  4A 11 66 0A 52 89 52 01  : F9
0A40  B2 3C 00 09 65 F2 4A 47  : DF
0A48  67 04 10 FC 00 2D 10 19  : CD
0A50  D0 3C 00 30 10 C0 52 01  : 5F
0A58  B2 3C 00 0A 65 F0 42 10  : 9F
0A60  4E 5E 4C DF 06 87 4E 75  : 27
0A68  3B 9A CA 00 05 F5 E1 00  : 7A
0A70  00 98 96 80 00 0F 42 40  : 3F
0A78  00 01 86 A0 00 00 27 10  : 5E
-----------------------------------
SUM:  65 4A AC E2 F3 DF 74 91  B45A

0A80  00 00 03 E8 00 00 00 64  : 4F
0A88  00 00 00 0A 00 00 00 01  : 0B
0A90  00 00 00 00 61 00 04 46  : AB
0A98  B0 3C 00 26 66 00 01 FE  : 77
0AA0  52 88 10 18 61 00 FE A0  : 01
0AA8  B0 3C 00 42 67 12 B0 3C  : 93
0AB0  00 4F 67 12 B0 3C 00 48  : FC
0AB8  67 12 44 FC 00 09 4E 75  : 85
0AC0  61 00 01 1A 60 0A 61 00  : 47
0AC8  00 9C 60 04 61 00 00 E0  : 41
0AD0  65 08 61 00 0E 26 44 FC  : 42
0AD8  00 00 4E 75 48 E7 7F 00  : 71
0AE0  4E 56 00 00 42 80 61 00  : C7
0AE8  03 F4 67 44 61 00 03 D8  : DE
0AF0  10 10 61 00 FE 02 65 38  : 1E
0AF8  90 3C 00 30 42 81 12 00  : D1
-----------------------------------
SUM:  D0 9B 96 87 39 71 00 2E  C0F1

0B00  52 88 10 10 61 00 FD F0  : 48
0B08  65 32 90 3C 00 30 2F 01  : C3
0B10  E3 89 65 10 E3 89 65 0C  : BE
0B18  D2 9F 65 08 E3 89 65 04  : B3
0B20  D2 80 64 DC 4E 5E 4C DF  : 69
0B28  00 FE 44 FC 00 03 4E 75  : 04
0B30  4E 5E 4C DF 00 FE 44 FC  : 15
0B38  00 09 4E 75 4A 47 6B 12  : DA
0B40  4A 81 6B E0 20 01 4E 5E  : E3
0B48  4C DF 00 FE 44 FC 00 00  : 69
0B50  4E 75 4A 81 6A 0A B2 BC  : 70
0B58  80 00 00 00 67 E6 60 C4  : F1
0B60  44 81 60 E0 2F 01 42 80  : F7
0B68  10 10 61 00 FD AA 65 36  : C3
0B70  90 3C 00 30 22 00 52 88  : F8
0B78  10 10 61 00 FD 9A 65 14  : 91
-----------------------------------
SUM:  E4 79 83 FF 3F 1A FD 93  D81C

0B80  E3 89 65 1A E3 89 65 16  : D2
0B88  E3 89 65 12 90 3C 00 30  : DF
0B90  82 00 60 E2 20 01 22 1F  : 26
0B98  44 FC 00 00 4E 75 22 1F  : 44
0BA0  44 FC 00 03 4E 75 22 1F  : 47
0BA8  44 FC 00 09 4E 75 2F 01  : 3C
0BB0  42 80 10 10 61 00 FD 70  : B0
0BB8  65 EC 61 00 FD 9E 22 00  : 6F
0BC0  52 88 10 10 61 00 FD 60  : B8
0BC8  65 CA B2 BC 10 00 00 00  : AD
0BD0  64 CC E9 89 61 00 FD 84  : 84
0BD8  82 00 60 E4 2F 01 42 80  : B8
0BE0  10 10 61 00 FD 22 65 BE  : C3
0BE8  90 3C 00 30 22 00 52 88  : F8
0BF0  10 10 61 00 FD 12 65 9C  : 91
0BF8  E3 89 65 A2 90 3C 00 30  : 6F
-----------------------------------
SUM:  EB 75 CD 35 88 34 71 8A  0D7D

0C00  82 00 60 EA 48 E7 E0 40  : 1B
0C08  4E 56 FF DE 22 4F 72 1F  : 83
0C10  74 30 E3 88 64 02 52 02  : C9
0C18  12 C2 51 C9 FF F4 60 5A  : 9B
0C20  48 E7 E0 40 4E 56 FF DE  : D0
0C28  22 4F 74 0A 42 01 60 06  : 98
0C30  42 01 E3 88 E3 11 E3 88  : 0D
0C38  E3 11 E3 88 E3 11 D2 3C  : 61
0C40  00 30 12 C1 51 CA FF EA  : 07
0C48  60 30 48 E7 E0 40 4E 56  : 83
0C50  FF DE 22 4F 74 07 42 01  : 0C
0C58  E3 88 E3 11 E3 88 E3 11  : BE
0C60  E3 88 E3 11 E3 88 E3 11  : BE
0C68  B2 3C 00 0A 65 02 5E 01  : BE
0C70  D2 3C 00 30 12 C1 51 CA  : 2C
0C78  FF DE 42 11 22 4F 0C 11  : BE
-----------------------------------
SUM:  8D 34 31 D7 27 D8 28 A2  6A27

0C80  00 30 66 0A 4A 29 00 01  : 14
0C88  67 04 52 89 60 F0 10 D9  : 7F
0C90  66 FC 53 88 4E 5E 4C DF  : 14
0C98  02 07 4E 75 48 E7 3F 60  : 9A
0CA0  4E 56 FF E2 61 00 02 36  : 1E
0CA8  67 00 00 E6 61 00 02 18  : C8
0CB0  3D 7C FF FF FF FA 42 AE  : A0
0CB8  FF FC 10 10 42 86 B0 3C  : CF
0CC0  00 2E 66 0C 08 C6 00 1F  : 8D
0CC8  42 6E FF FA 52 88 10 10  : A3
0CD0  61 00 FC 24 65 00 00 BA  : A0
0CD8  52 88 61 00 00 DC 43 EE  : 48
0CE0  FF E2 12 FC 00 30 24 49  : 8C
0CE8  12 C0 7A 01 10 10 B0 3C  : 59
0CF0  00 2E 66 12 4A 86 66 00  : DC
0CF8  00 98 08 C6 00 1F 42 6E  : 35
-----------------------------------
SUM:  C6 91 23 66 5C ED 60 1B  0A3F

0D00  FF FA 52 88 60 E6 61 00  : 7A
0D08  FB EE 65 14 12 C0 52 88  : 0E
0D10  61 00 00 A6 52 45 BA 7C  : D4
0D18  00 0F 66 D0 61 00 01 64  : 0B
0D20  61 00 00 F4 61 00 00 9A  : 50
0D28  4A 47 6A 04 61 00 0D CE  : 3B
0D30  2D 40 FF F2 2D 41 FF F6  : C1
0D38  4A 6E FF FA 67 24 61 00  : 9D
0D40  0C 16 65 1A 28 00 61 00  : 2A
0D48  0B B2 24 2E FF F2 26 2E  : 54
0D50  FF F6 61 00 0D C2 66 06  : 91
0D58  2D 44 FF FC 60 04 42 6E  : 80
0D60  FF FA 20 2E FF F2 22 2E  : 88
0D68  FF F6 42 82 34 2E FF FA  : 14
0D70  26 2E FF FC 4E 5E 50 8F  : DA
0D78  4C DF 06 F0 B0 BC 80 00  : 0D
-----------------------------------
SUM:  30 EB D5 D6 40 42 FB 1F  67EC

0D80  00 00 66 06 4A 81 66 02  : 9F
0D88  42 80 44 FC 00 00 4E 75  : C5
0D90  4E 5E 4C DF 06 FC 42 80  : 9B
0D98  42 81 44 FC 00 09 4E 75  : CF
0DA0  4E 5E 4C DF 06 FC 44 FC  : 19
0DA8  00 01 4E 75 4E 5E 4C DF  : 9B
0DB0  06 FC 44 FC 00 03 4E 75  : 08
0DB8  4A 86 6A 02 53 46 4E 75  : 98
0DC0  42 80 42 81 24 3C 40 24  : 49
0DC8  00 00 76 00 61 00 14 2E  : 19
0DD0  24 00 26 01 42 80 10 1A  : 37
0DD8  90 3C 00 30 61 00 0B 1C  : 84
0DE0  61 00 14 02 53 45 66 DC  : 51
0DE8  D8 46 67 28 6B 14 24 3C  : 8C
0DF0  40 24 00 00 76 00 61 00  : 3B
0DF8  14 04 65 B0 53 44 66 EE  : 18
-----------------------------------
SUM:  F3 6A 40 BB A6 82 30 BF  79ED

0E00  4E 75 24 3C 40 24 00 00  : 87
0E08  76 00 61 00 13 FC 65 90  : DB
```

```
0E10   52 44 66 EE 4E 75 42 84   : 73
0E18   B0 3C 00 45 67 08 B0 3C   : 8C
0E20   00 65 67 02 4E 75 42 6E   : 41
0E28   FF FA 52 88 10 10 B0 3C   : DF
0E30   00 2B 67 0A B0 3C 00 2D   : B5
0E38   66 08 08 C4 00 1F 52 88   : 33
0E40   10 10 61 00 FA B2 65 00   : 92
0E48   FF 48 90 3C 00 30 18 00   : 5B
0E50   42 40 52 88 10 10 61 00   : DD
0E58   FA 9E 65 1E 90 3C 00 30   : 17
0E60   3F 04 E5 4C D8 5F E3 4C   : DA
0E68   D8 40 B8 7C 03 E8 65 E2   : 7E
0E70   4A 84 6A 00 FF 38 60 00   : CF
0E78   FF 28 4A 84 6A 02 44 44   : E9
------------------------------------------
SUM:   D6 AD 0C F5 F4 2C 65 51   22AE

0E80   4E 75 52 46 53 45 0C 21   : 20
0E88   00 35 65 0E 52 21 0C 11   : 38
0E90   00 3A 65 06 12 BC 00 30   : A3
0E98   60 F2 B3 CA 64 04 24 49   : A4
0EA0   52 46 10 10 61 00 FA 50   : 63
0EA8   64 10 B0 3C 00 2E 66 14   : 08
0EB0   4A 86 6B 00 FE DC 08 C6   : E3
0EB8   00 1F 52 88 4A 86 6B E2   : 16
0EC0   52 46 60 DE 4E 75 42 47   : 22
0EC8   B0 3C 00 2B 67 08 B0 3C   : 72
0ED0   00 2D 66 04 46 47 52 88   : FE
0ED8   4E 75 52 88 10 10 B0 3C   : A9
0EE0   00 09 67 F6 B0 3C 00 20   : 72
0EE8   67 F0 4A 00 4E 75 44 80   : 28
0EF0   C4 7C 00 FF B0 42 63 02   : 96
0EF8   30 02 61 00 00 E6 10 FC   : 85
------------------------------------------
SUM:   59 6C 76 82 7D 63 BA 9C   F103

0F00   00 30 61 00 00 DE 10 FC   : 7B
0F08   00 2E 4A 40 67 0C 61 00   : 8C
0F10   00 D2 10 BC 00 30 53 40   : 61
0F18   66 F4 61 00 00 DE 4C DF   : C4
0F20   00 06 4E 75 48 E7 60 00   : 58
0F28   61 00 00 E2 4A 81 67 08   : 7D
0F30   61 00 00 B0 10 FC 00 2D   : 4A
0F38   4A 80 67 5A 6B 20 C4 7C   : 56
0F40   00 FF B0 42 67 60 64 36   : 52
0F48   41 F0 08 00 61 00 00 94   : 2E
0F50   10 BC 00 2E 61 00 00 A4   : FF
0F58   61 00 00 A6 60 48 2F 08   : E6
0F60   61 00 00 98 61 00 00 9A   : F4
0F68   91 D7 52 88 22 00 44 81   : 29
0F70   D2 88 20 5F C4 7C 00 FF   : 18
0F78   B2 42 65 00 FF 72 52 88   : A4
------------------------------------------
SUM:   9A F6 60 F2 43 12 C4 E4   CD07

0F80   61 60 10 BC 00 2E 61 72   : 8E
0F88   61 76 10 FC 00 45 53 80   : FB
0F90   61 1C 42 10 60 10 61 4A   : EA
0F98   10 FC 00 30 61 44 10 FC   : ED
0FA0   00 2E 61 56 61 5A 61 52   : 53
0FA8   4C DF 00 06 4E 75 12 3C   : 42
0FB0   00 2B 4A 80 6A 06 12 3C   : B3
0FB8   00 2D 44 80 10 C1 32 3C   : 30
0FC0   00 64 61 0E 32 3C 00 0A   : 4B
0FC8   61 08 D0 3C 00 30 10 C0   : 75
0FD0   4E 75 14 3C 00 2F 52 02   : 96
0FD8   90 41 64 FA D0 41 10 C2   : 12
0FE0   4E 75 48 E7 C0 80 10 18   : 5A
0FE8   12 10 10 C0 67 04 10 01   : 6E
0FF0   60 F6 4C DF 01 03 4E 75   : 48
0FF8   52 88 4A 10 66 FA 4E 75   : 57
------------------------------------------
SUM:   D0 78 E8 6A 7A BA 0A CF   ECB5

1000   0C 20 00 30 67 FA 52 88   : 97
1008   42 10 4E 75 48 E7 3F E0   : 63
1010   4E 56 FF EA 61 00 02 F0   : E0
1018   30 2E FF EE 61 00 03 18   : C7
1020   30 2E FF EE 61 00 03 2C   : DB
1028   4E 5E 30 07 48 C0 22 06   : 13
1030   4C DF 07 FC 4E 75 48 E7   : 20
1038   3F E0 4E 56 FF EA 61 00   : 0D
1040   02 C6 30 2E FF EE D0 47   : 2A
1048   61 00 02 EC 30 2E FF EE   : 9A
1050   D0 47 61 00 02 FE 4E 5E   : 24
1058   30 07 48 C0 22 06 4C DF   : 92
1060   07 FC 4E 75 48 E7 3F E0   : 14
1068   4A 83 6A 02 42 83 61 00   : 5F
1070   0A FE 67 06 08 04 00 03   : 84
1078   66 76 08 04 00 04 67 02   : 55
------------------------------------------
SUM:   F9 06 D2 1F 4C 92 D4 E0   7103

1080   52 82 48 E7 30 00 4E 56   : D7
1088   FF EA 24 03 61 00 02 78   : EB
1090   30 2E FF EE D0 47 61 00   : C3
1098   02 9E 30 2E FF EE D0 47   : 02
10A0   61 00 02 B0 61 00 01 36   : AB
10A8   61 00 01 56 61 00 01 6A   : 84
10B0   61 00 01 9E 61 1C 61 00   : DE
10B8   02 10 61 00 01 BA 61 00   : 8F
10C0   02 20 4E 5E 30 07 48 C0   : 0D
10C8   22 06 50 8F 4C DF 07 FC   : 35
10D0   4E 75 08 2E 00 03 00 17   : 13
10D8   67 14 20 6E FF EA 61 00   : 53
10E0   FF 1A 10 FC 00 45 42 80   : 2C
10E8   61 00 FE C4 42 10 4E 75   : 38
10F0   08 04 00 04 66 1A 08 04   : 9C
10F8   00 05 66 14 08 04 00 06   : 91
------------------------------------------
SUM:   E9 1A 3A 0B AF 51 8D 87   4376

1100   66 0E 4A 82 67 0A 53 82   : 86
1108   66 06 4A 83 66 02 74 01   : 16
```

```
1110   48 E7 30 00 4E 56 FF EA   : EC
1118   D4 83 61 00 01 EA 30 2E   : 01
1120   FF EE 61 00 02 12 30 2E   : C0
1128   FF EE 61 00 02 26 61 28   : FF
1130   20 6E FF EA 61 00 FE C4   : 9A
1138   10 FC 00 45 30 07 48 C0   : 90
1140   61 00 FE 6C 42 10 61 2E   : AC
1148   4E 5E 30 07 48 C0 22 06   : 13
1150   50 8F 4C DF 07 FC 4E 75   : D0
1158   20 2E 00 04 4A AE 00 10   : 5A
1160   6B 10 20 6E FF EA 41 F0   : 23
1168   00 00 61 00 FE 76 10 BC   : A1
1170   00 2E 9E 40 4E 75 20 6E   : 5D
1178   FF EA 20 2E 00 14 08 00   : 53
------------------------------------------
SUM:   9F 07 9F 66 D7 EE 17 48   6038

1180   00 04 66 28 08 00 00 05   : 9F
1188   66 40 08 00 00 06 66 32   : 4C
1190   4A 86 66 24 22 2E 00 0C   : B6
1198   67 40 70 20 B2 7C 00 02   : 67
11A0   64 18 4A AE 00 08 67 32   : 15
11A8   70 30 60 0E 4A 86 66 08   : 4C
11B0   70 2B 60 06 70 20 60 02   : F3
11B8   70 2D 61 00 FE 26 10 80   : B2
11C0   4E 75 61 00 FE 36 70 20   : E8
11C8   60 06 61 00 FE 2E 70 2B   : 8E
11D0   4A 86 67 02 70 2D 10 C0   : A6
11D8   42 10 4E 75 4A 47 6A 1E   : 2E
11E0   44 47 20 6E FF EA BE 6E   : 2E
11E8   FF EE 65 06 3E 2E FF EE   : B1
11F0   67 0C 61 00 FD EE 10 BC   : 8B
11F8   00 30 53 47 66 F4 4E 75   : E7
------------------------------------------
SUM:   AF 2C 5F 60 EA 56 18 B7   2FC5

1200   4A AE 00 10 6B 10 20 6E   : 11
1208   FF EA 41 F0 70 00 61 00   : EB
1210   FD D2 10 BC 00 2E 4E 75   : 8C
1218   4A 47 66 32 20 2E 00 0C   : 83
1220   67 2C 55 80 64 1A 20 2E   : 34
1228   00 14 08 00 00 04 66 10   : 96
1230   08 00 00 05 66 0A 08 00   : 85
1238   00 06 66 04 4A 86 66 0E   : B4
1240   20 6E FF EA 61 00 FD 9C   : 71
1248   10 BC 00 30 52 47 4E 75   : 58
1250   08 2E 00 02 00 17 67 1C   : D2
1258   20 6E FF EA 41 F0 70 00   : 18
1260   57 88 B1 EE FF EA 63 0C   : D6
1268   61 00 FD 78 10 BC 00 2C   : CE
1270   52 47 60 EC 4E 75 20 6E   : 36
1278   FF EA 20 2E 00 14 08 00   : 53
------------------------------------------
SUM:   60 76 A6 FD 60 97 70 0E   CE4A

1280   00 04 66 1C 08 00 00 05   : 93
1288   66 1E 08 00 00 06 66 24   : 1C
1290   4A 86 67 0A 70 2D 61 00   : 3F
1298   FD 4A 10 80 52 47 4E 75   : 33
12A0   4A 86 66 F0 70 2B 60 EE   : 0F
12A8   61 00 FD 50 4A 86 66 10   : F4
12B0   70 2B 60 0E 61 00 FD 44   : AB
12B8   4A 86 66 04 70 20 60 02   : 2C
12C0   70 2D 10 C0 42 10 4E 75   : 82
12C8   08 2E 00 01 00 17 67 0E   : C3
12D0   20 6E FF EA 61 00 FD 0C   : E1
12D8   10 BC 00 5C 52 47 4E 75   : 84
12E0   48 C7 9E AE 00 04 64 1C   : DF
12E8   20 6E FF EA 44 87 70 20   : D2
12F0   08 2E 00 00 00 17 67 02   : B6
12F8   70 2A 61 00 FC E6 10 80   : 6D
------------------------------------------
SUM:   9A 3B 1B 97 8A 41 83 A4   54E6

1300   53 87 66 F6 4E 75 2D 48   : 6E
1308   FF EA 42 6E FF EE 1D 42   : E5
1310   FF EF 61 6A 43 EE FF F0   : D9
1318   12 FC 00 30 2F 09 61 00   : D7
1320   00 FE 41 EE FF FF 61 00   : 8C
1328   01 1C 22 5F B1 C9 64 04   : 80
1330   22 48 52 47 4E 75 4A 40   : 50
1338   6B 16 B0 7C 00 0E 64 10   : 2F
1340   41 F1 00 00 61 00 00 FE   : 91
1348   B1 C9 64 04 22 48 52 47   : E5
1350   4E 75 20 6E FF EA 4A 40   : C4
1358   67 20 6B 1E 42 41 B2 7C   : C1
1360   00 FF 67 16 B2 40 67 12   : E7
1368   B2 7C 00 0E 64 06 10 D9   : 8F
1370   52 41 60 EA 10 FC 00 30   : 19
1378   60 F6 42 10 4E 75 42 86   : 33
------------------------------------------
SUM:   FC D5 66 BC F5 CF 24 70   1299

1380   24 00 C0 BC 7F FF FF FF   : 1C
1388   66 08 4A 81 66 04 42 47   : 2C
1390   4E 75 2E 00 CE BC 7F F0   : EA
1398   00 00 BE BC 7F F0 00 00   : E9
13A0   66 06 20 07 53 80 72 FF   : D7
13A8   E3 8A E3 96 7E 0F 24 3C   : D3
13B0   43 0C 6B F5 26 3C 26 34   : 6B
13B8   00 00 61 00 07 5A 64 24   : 4A
13C0   24 3C 3F F0 00 00 76 00   : 05
13C8   61 00 07 4C 64 2C 24 3C   : A4
13D0   43 0C 6B F5 26 3C 26 34   : 6B
13D8   00 00 61 00 0E 20 9E 7C   : A9
13E0   00 0F 60 CA 24 3C 43 0C   : E8
13E8   6B F5 26 3C 26 34 00 00   : 1C
13F0   61 00 0E 16 DE 7C 00 0F   : EE
13F8   60 B4 24 3C 42 D6 BC C4   : 0C
------------------------------------------
SUM:   58 19 8F 14 32 1E 3D 94   4C6F

1400   26 3C 1E 90 00 00 61 00   : 71
1408   07 0E 64 10 24 3C 40 24   : 4D
```

```
1410   00 00 76 00 61 00 0D E6   : CA
1418   53 47 60 DE 4E 75 38 3C   : 0F
1420   00 0E 45 FA 00 3E 12 BC   : 59
1428   00 30 41 D2 61 00 00 AC   : 50
1430   65 08 52 11 61 00 00 B8   : E9
1438   60 F2 52 89 50 8A 51 CC   : 24
1440   FF E6 4E 75 0C 10 00 35   : F9
1448   65 0E 52 20 0C 10 00 39   : 3A
1450   63 06 10 BC 00 30 60 F2   : B7
1458   4E 75 43 0C 6B F5 26 34   : CC
1460   00 00 42 D6 BC C4 1E 90   : 46
1468   00 00 42 A2 30 9C E5 40   : D5
1470   00 00 42 6D 1A 94 A2 00   : FF
1478   00 00 42 37 48 76 E8 00   : 1F
------------------------------------------
SUM:   5A 38 7D 5D B6 28 5C 96   B75D

1480   00 00 42 02 A0 5F 20 00   : 63
1488   00 00 41 CD CD 65 00 00   : 40
1490   00 00 41 97 D7 84 00 00   : 33
1498   00 00 41 63 12 D0 00 00   : 86
14A0   00 00 41 2E 84 80 00 00   : 73
14A8   00 00 40 F8 6A 00 00 00   : A2
14B0   00 00 40 C3 88 00 00 00   : 8B
14B8   00 00 40 8F 40 00 00 00   : 0F
14C0   00 00 40 59 00 00 00 00   : 99
14C8   00 00 40 24 00 00 00 00   : 64
14D0   00 00 3F F0 00 00 00 00   : 2F
14D8   00 00 24 10 26 28 00 04   : 86
14E0   60 00 06 34 24 10 26 28   : 1C
14E8   00 04 60 00 0B 9C 24 10   : 3F
14F0   26 28 00 04 60 00 0C FA   : B8
14F8   24 10 26 28 00 04 60 00   : E6
------------------------------------------
SUM:   AA 3C 75 1E C1 70 D6 36   372B

1500   0C A0 24 10 26 28 00 04   : 32
1508   60 00 0D 22 20 3C 40 09   : 34
1510   21 FB 22 3C 54 44 2D 18   : 57
1518   4E 75 48 E7 30 00 24 3C   : 82
1520   40 09 21 FB 26 3C 54 44   : 5F
1528   2D 18 61 00 0C D0 4C DF   : AD
1530   00 0C 4E 75 4A 80 6A 1C   : 1F
1538   B0 BC 80 00 00 00 66 04   : 56
1540   4A 81 67 0A 20 3C BF F0   : 47
1548   00 00 72 00 4E 75 42 80   : F7
1550   42 81 4E 75 66 04 4A 81   : BB
1558   67 08 20 3C 3F F0 00 00   : FA
1560   72 00 4E 75 48 E7 C0 00   : 24
1568   20 02 22 03 61 00 06 00   : AE
1570   67 24 61 00 03 1E 61 00   : 6E
1578   05 F6 67 38 4C DF 00 03   : C8
------------------------------------------
SUM:   E9 1F 6A 30 51 BD 73 98   253B

1580   61 00 05 EC 67 1C 61 00   : 36
1588   0D CE 65 14 61 00 0C 6E   : 2F
1590   65 0E 60 00 0D B6 50 8F   : 75
1598   20 3C 3F F0 00 00 72 00   : FD
15A0   4E 75 4A 82 6A FA 20 3C   : 4F
15A8   7F FF FF FF 72 FF 44 FC   : 2D
15B0   00 05 4E 75 4C DF 00 03   : F6
15B8   48 E7 31 00 4E 56 00 00   : 04
15C0   48 E7 C0 00 2F 02 08 82   : AA
15C8   00 1F 20 02 22 03 61 00   : C7
15D0   03 86 65 22 2E 00 61 48   : E7
15D8   65 1C 4A 9F 6A 10 24 00   : 08
15E0   26 01 20 3C 3F F0 00 00   : B2
15E8   72 00 61 00 0C 1C 4E 5E   : A7
15F0   4C DF 00 8C 4E 75 40 C0   : 7A
15F8   4A AE FF F4 6A 04 08 40   : A1
------------------------------------------
SUM:   E6 AE E0 65 37 9A 17 60   7CA6

1600   00 01 08 00 00 01 66 0A   : 7A
1608   42 80 42 81 44 FC 00 01   : C6
1610   60 DC 20 3C 7F FF FF FF   : 14
1618   72 FF 44 FC 00 03 60 CE   : E2
1620   4A 87 67 28 08 07 00 00   : 6F
1628   67 14 53 87 67 24 61 F0   : 31
1630   65 C4 24 2E FF F8 26 2E   : C6
1638   FF FC 60 00 0B C0 E2 8F   : 97
1640   61 DE 65 B2 24 00 26 01   : A1
1648   60 00 0B B2 42 80 42 81   : A2
1650   4E 75 20 2E FF F8 22 2E   : 58
1658   FF FC 4E 75 48 E7 71 00   : 5E
1660   2E 00 67 0A 4A 81 67 0C   : DD
1668   61 00 04 76 20 02 4C DF   : 28
1670   00 8E 4E 75 42 80 4C DF   : 3E
1678   00 8E 00 3C 00 01 4E 75   : 8E
------------------------------------------
SUM:   C6 22 83 CE 95 45 76 74   2BB7

1680   48 E7 40 00 4A 80 67 2E   : CE
1688   2F 00 02 97 80 00 00 00   : 48
1690   22 3C 00 00 00 9E 4A 80   : C6
1698   6A 02 44 80 6B 06 53 41   : 35
16A0   E3 88 60 F8 E8 99 EA 99   : C7
16A8   E3 88 E8 88 EA 88 08 80   : D5
16B0   00 1F 80 9F 80 81 4C DF   : 6A
16B8   00 02 4E 75 40 E7 2F 01   : 1C
16C0   22 00 EE 80 48 40 48 C0   : 20
16C8   C0 BC 80 00 00 FF E1 81   : 5D
16D0   08 81 00 1F 80 81 22 1F   : EA
16D8   46 DF 90 3C 00 7F 65 2C   : 01
16E0   B0 3C 00 1F 64 2A 48 E7   : C8
16E8   E0 00 42 81 12 00 24 00   : D9
16F0   08 C2 00 1F 42 80 42 02   : EF
16F8   E3 8A E3 90 51 C9 FF FA   : F3
------------------------------------------
SUM:   74 FA BF D5 98 5F CE 57   6139

1700   4A 9F 6A 02 44 80 4C DF   : 44
1708   00 06 4E 75 42 80 4E 75   : 4E
```

▶乱数でメロディを作るのによく似たアイデアであらかじめ作っておいた曲の断片を指揮者のインスピレーションでつないでいくというものがあります。あのワルツ王ヨハン・シュトラウスの常動曲という曲です（昔「オーケストラがやってきた」のテーマ曲で使われた)。

属　真人（24）京都府

```
1710  66 10 4A 80 6A 0C 42 00  : F8
1718  B0 BC 80 00 00 00 66 02  : 54
1720  4E 75 20 3C 7F FF FF FF  : 9B
1728  44 FC 00 03 4E 75 48 E7  : 35
1730  20 00 24 00 C4 BC 7F F0  : 33
1738  00 00 67 40 B4 BC 47 F0  : 4E
1740  00 00 64 44 B4 BC 38 10  : 60
1748  00 00 65 30 4A 80 6B 04  : CE
1750  61 14 60 0C 08 80 00 1F  : 88
1758  61 0C 80 BC 80 00 00 00  : 29
1760  4C DF 00 04 4E 75 90 BC  : 3E
1768  38 00 00 00 34 3C 00 02  : AA
1770  E3 89 E3 90 51 CA FF FA  : F3
1778  42 81 4E 75 42 81 42 80  : 0B
-----------------------------------
SUM:  7D EB 07 BB D0 B0 C3 87  EC80

1780  60 DE 4C DF 00 04 4E 75  : 30
1788  42 81 20 3C 7F FF FF FF  : 9B
1790  44 FC 00 03 4C DF 00 04  : 72
1798  4E 75 42 81 4A 80 67 2A  : E1
17A0  6A 10 B0 BC 80 00 00 00  : 66
17A8  67 20 61 06 08 C0 00 1F  : D5
17B0  4E 75 08 80 00 1F 32 3C  : D8
17B8  00 07 E2 88 E2 91 51 C9  : FE
17C0  FF FA D0 BC 38 00 00 00  : BD
17C8  42 41 4E 75 48 E7 30 00  : A5
17D0  61 06 4C DF 00 0C 4E 75  : 61
17D8  24 00 ED 9A ED 9A C4 BC  : B2
17E0  00 00 07 FF 94 7C 03 FF  : 18
17E8  65 34 B4 7C 00 34 64 32  : 93
17F0  B4 7C 00 14 62 12 42 81  : 7B
17F8  26 3C 00 10 00 00 E4 AB  : 01
-----------------------------------
SUM:  58 A9 BB B2 E2 21 06 54  AB46

1800  53 83 46 83 C0 83 4E 75  : A5
1808  94 7C 00 15 26 3C 80 00  : 07
1810  00 00 E4 AB 53 83 46 83  : 2E
1818  C2 83 74 14 60 DA 42 80  : C9
1820  42 81 4E 75 48 E7 30 00  : E5
1828  4A 80 6A A4 24 00 26 01  : 23
1830  61 60 61 00 03 3A C5 40  : 64
1838  C7 41 67 0E 61 9A 24 3C  : D8
1840  3F F0 00 00 76 00 61 00  : 06
1848  09 A8 4C DF 00 0C 4E 75  : AB
1850  48 E7 30 00 4A 80 6B 00  : 94
1858  FF 78 24 00 26 01 61 32  : 55
1860  61 00 03 0C C5 40 C7 41  : 7D
1868  67 22 24 00 ED 9A ED 9A  : BB
1870  C4 BC 00 00 07 FF B4 7C  : B6
1878  04 33 64 10 61 00 FF 60  : 6B
-----------------------------------
SUM:  7C 2C 49 79 69 3D 77 53  1AFE

1880  24 3C 3F F0 00 00 76 00  : 05
1888  61 00 09 5A 4C DF 00 0C  : FB
1890  4E 75 48 E7 30 00 24 00  : 46
1898  26 00 ED 9A ED 9A C4 BC  : B4
18A0  00 00 07 FF B4 7C 03 FF  : 38
18A8  65 3C B4 7C 04 33 64 40  : AC
18B0  C0 BC 00 0F FF FF E3 89  : F5
18B8  E3 90 53 42 B4 7C 03 FF  : 3A
18C0  64 F4 08 00 00 14 66 0A  : E4
18C8  4A 80 66 EA 4A 81 66 E6  : 31
18D0  60 1E C0 BC 00 0F FF FF  : 07
18D8  4A 83 6A 04 84 7C 08 00  : 43
18E0  EC 9A EC 9A 80 82 02 3C  : 4C
18E8  00 FE 4C DF 00 0C 4E 75  : F8
18F0  42 80 42 81 4C DF 00 0C  : BC
18F8  4E 75 4A 80 67 54 48 E7  : 77
-----------------------------------
SUM:  D5 DB E7 BB D5 84 16 22  40CF

1900  30 00 42 81 26 00 C6 BC  : 9B
1908  80 00 00 00 4A 80 6A 02  : B6
1910  44 80 74 0B 4A 82 67 08  : 7E
1918  E2 88 E2 91 53 82 60 F4  : 06
1920  34 3C 04 1E C0 BC 00 1F  : 2D
1928  FF FF 08 00 00 14 66 0E  : 8E
1930  E3 89 E3 90 53 42 60 F2  : C6
1938  C4 BC 00 00 07 FF C0 BC  : 02
1940  00 0F FF FF EC 9A EC 9A  : 19
1948  80 82 80 83 4C DF 00 0C  : 3C
1950  4E 75 42 81 4E 75 48 E7  : 78
1958  68 00 24 00 ED 9A ED 9A  : 9A
1960  C4 7C 07 FF 94 7C 03 FF  : 58
1968  65 36 B4 7C 00 1F 64 38  : 86
1970  48 E7 90 00 26 00 42 80  : A7
1978  38 3C 00 0A E3 89 E3 93  : 60
-----------------------------------
SUM:  8F 63 B7 53 37 41 2A 06  A682

1980  51 CC FF FA 08 C3 00 1F  : 00
1988  E3 8B E3 90 51 CA FF FA  : F5
1990  4A 9F 6A 02 44 80 02 3C  : 57
1998  00 FE 4C DF 00 08 60 02  : 93
19A0  42 80 4C DF 00 16 4E 75  : C6
19A8  66 1C 4A 80 6A 18 B0 BC  : 3A
19B0  C1 E0 00 00 66 10 C2 BC  : 95
19B8  FF E0 00 00 66 08 20 3C  : A9
19C0  80 00 00 00 60 DC 4C DF  : E7
19C8  00 16 44 FC 00 03 4E 75  : 1C
19D0  48 E7 61 00 2E 00 67 3C  : 61
19D8  6A 02 44 80 B3 87 4A 81  : 35
19E0  67 32 6A 02 44 81 61 42  : 6D
19E8  4A 80 66 1E 20 01 4A 87  : 40
19F0  6A 14 B0 BC 80 00 00 00  : 6A
19F8  62 10 44 80 02 3C 00 FE  : 72
-----------------------------------
SUM:  95 25 DB A2 FA 7F 37 58  6DCB

1A00  4C DF 00 86 4E 75 4A 80  : 3E
1A08  6A F6 4C DF 00 86 44 FC  : 51
1A10  00 03 4E 75 42 80 4C DF  : B3
1A18  00 86 4E 75 48 E7 61 00  : D9
1A20  61 08 4A 80 66 E4 20 01  : 9E
1A28  60 D6 2F 02 34 01 C4 C0  : 20
1A30  2F 02 42 A7 24 00 48 42  : C8
1A38  C4 C1 D5 AF 00 02 64 02  : 71
1A40  52 57 24 01 48 42 C4 C0  : DC
1A48  D5 AF 00 02 64 02 52 57  : 95
1A50  48 40 48 41 C0 C1 D0 9F  : 01
1A58  22 1F 24 1F 4E 75 48 E7  : 76
1A60  71 00 2E 00 67 20 6A 02  : 92
1A68  44 80 B3 87 4A 81 67 2A  : 5A
1A70  6A 02 44 81 61 6A 4A 87  : CD
1A78  6A 0C 44 80 02 3C 00 FE  : 76
-----------------------------------
SUM:  84 F2 71 12 64 0A 14 AE  B737

1A80  4C DF 00 8E 4E 75 B0 BC  : E8
1A88  80 00 00 00 66 EE 53 80  : A7
1A90  44 FC 00 03 4C DF 00 8E  : FC
1A98  4E 75 4C DF 00 8E 00 3C  : B8
1AA0  00 01 4E 75 48 E7 71 00  : 64
1AA8  2E 00 67 DA 6A 02 44 80  : 9F
1AB0  4A 81 67 E6 6A 02 44 81  : 49
1AB8  61 26 4A 87 6A 02 44 82  : 8A
1AC0  20 02 4C DF 00 8E 4E 75  : 9E
1AC8  48 E7 71 00 4A 80 67 0A  : DB
1AD0  4A 81 67 C6 61 0A 02 3C  : A1
1AD8  00 FE 4C DF 00 8E 4E 75  : 7A
1AE0  2F 03 42 82 76 1F E3 88  : F6
1AE8  E3 92 B4 81 65 06 08 C0  : DD
1AF0  00 00 94 81 51 CB FF F0  : 20
1AF8  26 1F 4E 75 2F 00 02 9F  : D8
-----------------------------------
SUM:  21 14 FA A9 8C 53 31 90  48CD

1B00  7F FF FF FF 66 04 4A 81  : B1
1B08  67 04 08 40 00 1F 4E 75  : 95
1B10  08 80 00 1F 4E 75 48 E7  : 99
1B18  A0 00 4A 81 66 0A B0 BC  : 47
1B20  80 00 00 00 66 02 42 80  : AA
1B28  4A 83 66 0A B4 BC 80 00  : 2D
1B30  00 00 66 02 42 82 61 06  : 93
1B38  4C DF 00 05 4E 75 2F 00  : 22
1B40  B5 9F 6B 0E 4A 80 6A 1E  : 1F
1B48  61 1C 67 20 0A 3C 00 F9  : 43
1B50  4E 75 2F 00 E3 D7 58 8F  : 93
1B58  64 06 44 FC 00 F9 4E 75  : 66
1B60  02 3C 00 00 4E 75 B0 82  : 33
1B68  66 02 B2 83 4E 75 4A 80  : 2A
1B70  6B 0A 66 06 4A 81 02 3C  : EA
1B78  00 17 4E 75 48 E7 80 00  : 89
-----------------------------------
SUM:  3F 7A C8 18 29 35 6E 78  21ED

1B80  02 9F 7F FF FF FF 66 04  : 87
1B88  4A 81 67 EE 00 3C 00 08  : 64
1B90  4E 75 20 3C 69 65 65 65  : B7
1B98  22 3C 66 70 63 70 4E 75  : CA
1BA0  D0 BC 00 00 80 00 B0 BC  : 78
1BA8  00 01 00 00 65 04 70 FF  : D9
1BB0  4E 75 48 E7 40 00 C0 BC  : AE
1BB8  00 00 FF FF E5 88 72 01  : DE
1BC0  08 80 00 02 67 02 72 03  : 68
1BC8  80 81 23 C0 00 00 1C 06  : 06
1BD0  4C DF 00 02 42 80 4E 75  : B2
1BD8  61 24 48 E7 30 00 61 00  : 45
1BE0  FD 1A 24 3C 40 F0 00 00  : A7
1BE8  76 00 61 00 06 1C 4C DF  : 24
1BF0  00 0C 4E 75 61 08 C0 BC  : B4
1BF8  00 00 7F FF 4E 75 48 E7  : 70
-----------------------------------
SUM:  82 2D 70 DA A3 A7 FC 5E  2B91

1C00  40 80 41 FA 00 42 2F 10  : 7C
1C08  20 10 22 3C 00 00 03 83  : 14
1C10  61 00 FE 0A C0 BC 00 03  : E8
1C18  FF FF 20 80 20 1F 42 81  : A0
1C20  E2 88 E2 88 D1 41 4C DF  : 11
1C28  01 02 4E 75 61 D0 48 E7  : 26
1C30  40 00 61 00 FA 4C 22 3C  : 45
1C38  47 80 00 00 61 00 08 A6  : D6
1C40  4C DF 00 02 4E 75 00 00  : F0
1C48  00 01 61 00 03 A6 61 00  : 6C
1C50  F3 BC 4E 75 61 00 03 9C  : 72
1C58  61 00 F3 DC 4E 75 61 00  : 54
1C60  03 92 61 00 F2 C0 4E 75  : 6B
1C68  48 E7 40 00 61 00 03 84  : 57
1C70  61 00 F3 F2 61 00 02 B2  : 5B
1C78  4C DF 00 02 4E 75 48 E7  : 1F
-----------------------------------
SUM:  C2 8D 48 04 6F 3F 92 ED  1BBC

1C80  40 00 61 00 EE 10 61 00  : 00
1C88  02 A0 4C DF 00 02 4E 75  : 92
1C90  48 E7 40 00 61 00 F0 06  : C6
1C98  61 00 02 8E 4C DF 00 02  : 1E
1CA0  4E 75 2F 01 61 00 03 4C  : A3
1CA8  61 00 EC C2 61 00 02 7A  : EC
1CB0  22 1F 4E 75 4A 80 6A 10  : 48
1CB8  48 E7 80 00 02 9F 7F FF  : CE
1CC0  FF FF 67 04 00 3C 00 08  : AD
1CC8  4E 75 48 E7 C0 00 B0 BC  : 1E
1CD0  80 00 00 00 66 02 42 80  : AA
1CD8  B2 BC 80 00 00 00 66 02  : 56
1CE0  42 81 61 06 4C DF 00 03  : 58
1CE8  4E 75 2F 00 B3 9F 6B 0E  : BD
1CF0  4A 80 6A 1E 61 1C 67 1C  : 52
1CF8  0A 3C 00 F9 4E 75 2F 00  : 31
-----------------------------------
SUM:  67 E4 01 AD 7D 5D E6 C5  6803

1D00  E3 D7 58 8F 64 06 44 FC  : 4B
1D08  00 F9 4E 75 02 3C 00 00  : FA
1D10  4E 75 B0 81 4E 75 2F 00  : E6
1D18  02 9F 7F FF FF FF 67 04  : 88
1D20  08 40 00 1F 4E 75 08 80  : B2
1D28  00 1F 4E 75 48 E7 60 00  : 71
1D30  61 06 4C DF 00 06 4E 75  : 5B
1D38  22 00 E9 99 EB 99 C2 BC  : A6
1D40  00 00 00 FF 92 3C 00 7F  : 4C
1D48  65 20 B2 3C 00 17 64 1C  : 0A
1D50  24 3C 00 80 00 00 2F 00  : 0F
1D58  02 97 7F 80 00 00 E2 AA  : 24
1D60  53 82 46 82 C0 82 80 9F  : FE
1D68  4E 75 42 80 4E 75 48 E7  : 77
1D70  60 00 4A 80 6B BA 24 00  : 73
1D78  2F 02 61 BC 24 1F B4 80  : C5
-----------------------------------
SUM:  79 35 BC 09 63 D4 67 FC  B309

1D80  67 04 61 00 08 04 4C DF  : 03
1D88  00 06 4E 75 48 E7 60 00  : 58
1D90  4A 80 6A 9C 22 00 61 14  : 67
1D98  61 00 FF 1A C3 40 67 06  : EA
1DA0  61 96 61 00 07 F8 4C DF  : 82
1DA8  00 06 4E 75 48 E7 60 00  : 58
1DB0  22 00 24 00 EB 99 E9 99  : 4C
1DB8  C2 BC 00 00 00 FF B2 3C  : 6B
1DC0  00 7F 65 36 B2 3C 00 96  : 9E
1DC8  64 3A C0 BC 00 7F FF FF  : 97
1DD0  E3 88 53 41 B2 3C 00 7F  : 6C
1DD8  64 F6 08 00 00 17 66 06  : E5
1DE0  4A 80 66 EC 60 1E C0 BC  : 16
1DE8  00 7F FF FF 4A 82 6A 04  : B7
1DF0  82 7C 01 00 E8 99 EA 99  : 03
1DF8  80 81 02 3C 00 FE 4C DF  : 68
-----------------------------------
SUM:  4E 15 D3 FA 65 E7 80 FF  DB1E

1E00  00 06 4E 75 42 80 4C DF  : B6
1E08  00 06 4E 75 4A 80 6A 14  : 11
1E10  B0 BC 80 00 00 00 67 08  : 5B
1E18  20 3C BF 80 00 00 4E 75  : 5E
1E20  42 80 4E 75 4A 80 67 06  : BC
1E28  20 3C 3F 80 00 00 4E 75  : DE
1E30  20 3C 40 49 0F DA 4E 75  : 91
1E38  48 E7 40 00 22 3C 40 49  : 56
1E40  0F DA 61 00 06 94 4C DF  : 0F
1E48  00 02 4E 75 48 E7 80 00  : 74
1E50  20 01 61 00 FE 60 67 24  : 6B
1E58  61 00 FF 52 61 00 FE 56  : 67
1E60  67 34 4C DF 00 01 61 00  : 28
1E68  FE 4C 67 1A 61 00 07 B2  : E5
1E70  65 12 61 00 06 64 65 0C  : B3
1E78  60 00 07 9A 58 8F 20 3C  : 44
-----------------------------------
SUM:  54 52 12 02 73 65 CC FC  4F8E

1E80  3F 80 00 00 4E 75 4A 81  : 4D
1E88  6A FA 20 3C 7F FF FF FF  : 3C
1E90  44 FC 00 05 4E 75 4C DF  : 33
1E98  00 01 48 E7 21 00 4E 56  : F5
1EA0  00 00 48 E7 80 00 2F 01  : DF
1EA8  08 81 00 1F 20 01 61 00  : 2A
1EB0  F8 0C 65 1E 2E 00 61 40  : 56
1EB8  65 18 4A 9F 6A 0C 22 00  : FE
1EC0  20 3C 3F 80 00 00 61 00  : 7C
1EC8  06 1C 4E 5E 4C DF 00 82  : 7B
1ED0  4E 75 40 C0 4A AE FF F8  : B2
1ED8  6A 04 08 40 00 01 08 00  : BF
1EE0  00 01 66 08 42 80 44 FC  : 71
1EE8  00 01 60 DE 20 3C 7F FF  : 19
1EF0  FF FF 44 FC 00 03 60 D2  : 73
1EF8  4A 87 67 22 08 07 00 00  : 69
-----------------------------------
SUM:  79 75 A5 CD 74 4A 81 3D  CD19

1F00  67 10 53 87 67 1C 61 F0  : 25
1F08  65 C8 22 2E FF FC 60 00  : D8
1F10  05 C8 E2 8F 61 E2 65 BA  : A0
1F18  22 00 60 00 05 BC 42 80  : 05
1F20  4E 75 20 2E FF FC 4E 75  : CF
1F28  48 E7 20 00 40 E7 65 52  : 2D
1F30  24 00 C4 BC 7F F0 00 00  : 13
1F38  67 00 00 8A B4 BC 47 F0  : 98
1F40  00 00 64 00 00 8C B4 BC  : 60
1F48  38 10 00 00 65 00 00 96  : 43
1F50  4A 80 6B 04 61 58 60 0C  : 5E
1F58  08 80 00 1F 61 50 80 BC  : 94
1F60  80 00 00 00 24 00 08 82  : 2E
1F68  00 1F B4 BC 7F 80 00 00  : 8E
1F70  64 5E B4 BC 00 7F FF FF  : AF
1F78  65 6A 46 DF 4C DF 00 04  : 23
-----------------------------------
SUM:  E7 F3 38 32 54 57 FD 80  0B49

1F80  4E 75 67 0C 6B 1C 69 48  : 6E
1F88  61 00 FB E4 67 56 60 A0  : FD
1F90  46 DF 4C DF 00 04 20 3C  : B0
1F98  7F FF FF FF 44 FC 00 05  : C1
1FA0  4E 75 46 DF 4C DF 00 04  : 17
1FA8  44 FC 00 09 4E 75 90 BC  : 58
1FB0  38 00 00 00 34 3C 00 02  : AA
1FB8  E3 89 E3 90 51 CA FF FA  : F3
1FC0  42 81 4E 75 46 DF 42 81  : 6E
1FC8  42 80 4C DF 00 04 4E 75  : B4
1FD0  46 DF 42 81 20 3C 7F FF  : C2
1FD8  FF FF 44 FC 00 03 4C DF  : 6C
1FE0  00 04 4E 75 46 DF 42 80  : AE
1FE8  44 FC 00 01 4C DF 00 04  : 70
1FF0  4E 75 B0 BC 80 00 00 00  : AF
1FF8  67 26 4A 80 67 22 6A 08  : 52
-----------------------------------
SUM:  E3 C7 3E C9 14 CE 7F 45  B7FA

2000  61 06 08 C0 00 1F 4E 75  : 11
2008  08 80 00 1F 72 07 E2 88  : 8A
```

▶シャープが14型の液晶カラーディスプレイを開発したと新聞に載っていました。これを内蔵したラップトップパソコンがまた「世界初」というかたちでシャープから発表されるのもそう遠くないのかもしれません。
中山　剛（19）長崎県

```
2010   E2 91 51 C9 FF FA D0 BC   : 12
2018   38 00 00 00 42 41 4E 75   : 7E
2020   42 80 42 81 4E 75 4A 96   : 28
2028   67 2E 48 E7 06 E0 4C F9   : EF
2030   07 60 00 00 26 54 32 BC   : CF
2038   40 00 4A 50 24 96 4A 50   : 2E
2040   32 86 BA 50 67 FC 2C 92   : E3
2048   2D 52 00 04 4A 50 4C DF   : 48
2050   07 60 44 FC 00 00 4E 75   : 6A
2058   42 AE 00 04 60 F4 48 E7   : 77
2060   06 E0 4C F9 07 60 00 00   : 92
2068   26 54 32 BC 54 03 4A 50   : 59
2070   24 96 24 AE 00 04 4A 50   : 2A
2078   32 BC 60 00 BA 50 67 FC   : BB
-----------------------------------------
SUM:   9D 91 2D 17 77 97 69 32   AA41

2080   2C 92 42 AE 00 04 4A 50   : 4C
2088   48 46 32 86 4A 50 2C 12   : 1E
2090   08 06 00 0D 4A 50 66 0A   : 25
2098   4C DF 07 60 44 FC 00 00   : D2
20A0   4E 75 42 86 32 BC 88 00   : 01
20A8   4A 50 24 86 4C DF 07 60   : D6
20B0   44 FC 00 03 4E 75 48 E7   : 35
20B8   0E E0 28 3C 54 22 54 00   : 1C
20C0   60 62 48 E7 0E E0 28 3C   : 43
20C8   54 28 54 00 60 56 48 E7   : B5
20D0   0E E0 28 3C 54 23 54 00   : 1D
20D8   60 4A 48 E7 0E E0 4C EE   : 01
20E0   00 30 00 08 08 84 00 1F   : E3
20E8   8A 84 67 08 28 3C 54 20   : 55
20F0   54 00 60 30 2C BC 7F FF   : 4A
20F8   FF FF 2D 7C FF FF FF FF   : A3
-----------------------------------------
SUM:   B1 C5 09 B2 23 86 E9 01   A887

2100   00 04 4C DF 07 70 44 FC   : E6
2108   00 05 4E 75 48 E7 0E E0   : E5
2110   4C EE 00 30 00 08 08 84   : FE
2118   00 1F 8A 84 67 D6 28 3C   : CE
2120   54 21 54 00 4C F9 07 60   : 75
2128   00 00 26 54 32 84 4A 50   : CA
2130   24 96 24 AE 00 04 4A 50   : 2A
2138   48 44 32 84 BA 50 67 FC   : AF
2140   24 AE 00 08 24 AE 00 0C   : B8
2148   4A 50 32 86 BA 50 67 FC   : BF
2150   2C 92 2D 52 00 04 4A 50   : DB
2158   48 46 32 86 4A 50 2C 12   : 1E
2160   4A 50 4A 46 67 18 08 06   : B7
2168   00 0B 66 1C 08 06 00 07   : A2
2170   66 28 08 06 00 06 66 1C   : 24
2178   42 84 60 00 05 06 4C DF   : 5C
-----------------------------------------
SUM:   E0 EE 9D 5C 8A 82 1B 0A   67A1

2180   07 70 44 FC 00 00 4E 75   : 7A
2188   42 96 42 AE 00 04 78 01   : 45
2190   60 00 04 F0 78 03 60 00   : 2F
2198   04 EA 78 01 60 00 04 E4   : AF
21A0   48 E7 0E E0 28 3C 54 22   : F7
21A8   54 00 60 0A 48 E7 0E E0   : DB
21B0   28 3C 54 28 54 00 4C F9   : 79
21B8   07 60 00 00 26 54 32 84   : 97
21C0   4A 50 24 96 24 AE 00 04   : 2A
21C8   4A 50 48 44 32 84 BA 50   : E6
21D0   67 FC 24 BC 3F F0 00 00   : 72
21D8   24 BC 00 00 00 00 4A 50   : 7A
21E0   60 00 FF 68 48 E7 0E E0   : E4
21E8   28 3C 54 22 54 00 60 54   : E2
21F0   48 E7 0E E0 28 3C 54 28   : FD
21F8   54 00 60 48 48 E7 0E E0   : 19
-----------------------------------------
SUM:   BB EE 15 F5 63 AA DE B9   529B

2200   28 3C 54 23 54 00 60 3C   : CB
2208   48 E7 0E E0 28 02 08 84   : D3
2210   00 1F 88 83 67 08 28 3C   : FD
2218   54 20 54 00 60 26 20 3C   : AA
2220   7F FF FF FF 72 FF 4C DF   : 18
2228   07 70 44 FC 00 05 4E 75   : 7F
2230   48 E7 0E E0 28 02 08 84   : D3
2238   00 1F 88 83 67 E0 28 3C   : D5
2240   54 21 54 00 4C F9 07 60   : 75
2248   00 00 26 54 32 84 4A 50   : CA
2250   24 80 24 81 4A 50 48 44   : 6F
2258   32 84 BA 50 67 FC 24 82   : C9
2260   24 83 4A 50 32 86 BA 50   : 03
2268   67 FC 20 12 22 12 4A 50   : 63
2270   48 46 32 86 4A 50 2C 12   : 1E
2278   4A 50 4A 46 67 18 08 06   : B7
-----------------------------------------
SUM:   59 11 55 37 78 DF 6F 7A   BEEA

2280   00 0B 66 1C 08 06 00 07   : A2
2288   66 26 08 06 00 06 66 1A   : 20
2290   42 84 60 00 03 EE 4C DF   : 42
2298   07 70 44 FC 00 00 4E 75   : 7A
22A0   42 80 42 81 78 01 60 00   : 5E
22A8   03 DA 78 03 60 00 03 D4   : 8F
22B0   78 01 60 00 03 CE 48 E7   : D9
22B8   0E E0 28 3C 54 22 54 00   : 1C
22C0   2F 07 2E 3C 3F F0 00 00   : CF
22C8   60 26 48 E7 0E E0 28 3C   : 07
22D0   54 28 54 00 2F 07 2E 3C   : 70
22D8   3F F0 00 00 60 12 48 E7   : D0
22E0   0E E0 28 3C 54 20 54 00   : 1A
22E8   2F 07 2E 3C 40 00 00 00   : E0
22F0   4C F9 07 60 00 00 26 54   : 26
22F8   32 84 4A 50 24 80 24 81   : 99
-----------------------------------------
SUM:   57 09 C5 29 CE 74 3B 64   F1CC

2300   4A 50 48 44 32 84 BA 50   : E6
2308   67 FC 24 87 24 BC 00 00   : EE
2310   00 00 4A 50 2E 1F 60 00   : 47
2318   FF 4C 48 E7 0E E0 28 3C   : CC
2320   00 00 54 0E 60 78 48 E7   : 69
2328   0E E0 28 3C 00 00 54 1D   : C3
2330   60 6C 48 E7 0E E0 28 3C   : 4D
2338   00 00 54 0F 60 60 48 E7   : 52
2340   0E E0 28 3C 00 00 54 0A   : B0
2348   60 54 48 E7 0E E0 28 3C   : 35
2350   00 00 54 10 60 48 61 00   : 6D
2358   F8 16 6B 0E 67 16 48 E7   : 33
2360   0E E0 28 3C 00 00 54 14   : BA
2368   60 34 42 80 42 81 44 FC   : 59
2370   00 01 4E 75 44 FC 00 05   : 09
2378   4E 75 61 00 F7 F2 6B 0E   : 86
-----------------------------------------
SUM:   40 B8 5E B4 B2 A4 76 03   31C1

2380   67 16 48 E7 0E E0 28 3C   : FE
2388   00 00 54 04 60 10 42 80   : 8A
2390   42 81 44 FC 00 01 4E 75   : C7
2398   44 FC 00 00 4E 75 4C F9   : 48
23A0   07 60 00 00 26 54 32 84   : 97
23A8   4A 50 24 80 24 81 4A 50   : 7D
23B0   60 00 FE B2 48 E7 0E E0   : 2D
23B8   28 3C 44 22 44 00 60 5A   : C8
23C0   48 E7 0E E0 28 3C 44 28   : ED
23C8   44 00 60 4E 48 E7 0E E0   : 0F
23D0   28 3C 44 23 44 00 60 42   : B1
23D8   48 E7 0E E0 4C EE 00 10   : 67
23E0   00 04 02 84 7F FF FF FF   : 06
23E8   67 08 28 3C 44 20 44 00   : 7B
23F0   60 28 2C BC 7F FF FF FF   : EC
23F8   4C DF 07 70 44 FC 00 05   : E7
-----------------------------------------
SUM:   D5 9C 63 58 18 4D E2 95   ADD6

2400   4E 75 48 E7 0E E0 4C EE   : 1A
2408   00 10 00 04 02 84 7F FF   : 18
2410   FF FF 67 DE 28 3C 44 21   : 0C
2418   44 00 4C F9 07 60 00 00   : F0
2420   26 54 32 84 4A 50 24 96   : 84
2428   4A 50 48 44 32 84 BA 50   : E6
2430   67 FC 24 AE 00 04 4A 50   : D3
2438   32 86 BA 50 67 FC 2C 92   : E3
2440   4A 50 48 46 32 86 4A 50   : 7A
2448   2C 12 4A 50 4A 46 67 18   : E7
2450   08 06 00 0B 66 1C 08 06   : A9
2458   00 07 66 24 08 06 00 06   : A5
2460   66 18 42 84 60 00 02 1C   : C2
2468   4C DF 07 70 44 FC 00 00   : E2
2470   4E 75 42 96 78 01 60 00   : 74
2478   02 0A 78 03 60 00 02 04   : ED
-----------------------------------------
SUM:   1A 8F 4E DA 88 BF 80 6A   9ED2

2480   78 01 60 00 01 FE 48 E7   : 07
2488   0E E0 28 3C 44 22 44 00   : FC
2490   60 0A 48 E7 0E E0 28 3C   : EB
2498   44 28 44 00 4C F9 07 60   : 5C
24A0   00 00 26 54 32 84 4A 50   : CA
24A8   24 96 4A 50 48 44 32 84   : 96
24B0   BA 50 67 FC 24 BC 3F 80   : 0C
24B8   00 00 4A 50 60 00 FF 7A   : 73
24C0   48 E7 0E E0 28 3C 44 22   : E7
24C8   44 00 60 52 48 E7 0E E0   : 13
24D0   28 3C 44 28 44 00 60 46   : BA
24D8   48 E7 0E E0 28 3C 44 23   : E8
24E0   44 00 60 3A 48 E7 0E E0   : FB
24E8   28 01 02 84 7F FF FF FF   : 2B
24F0   67 08 28 3C 44 20 44 00   : 7B
24F8   60 24 20 3C 7F FF FF FF   : 5C
-----------------------------------------
SUM:   37 30 9F 83 03 E1 BB 9A   C065

2500   4C DF 07 70 44 FC 00 05   : E7
2508   4E 75 48 E7 0E E0 28 01   : 09
2510   02 84 7F FF FF FF 67 E2   : 4B
2518   28 3C 44 21 44 00 4C F9   : 52
2520   07 60 00 00 26 54 32 84   : 97
2528   4A 50 24 80 4A 50 48 44   : 64
2530   32 84 BA 50 67 FC 24 81   : C8
2538   4A 50 32 86 BA 50 67 FC   : BF
2540   20 12 4A 50 48 46 32 86   : 12
2548   4A 50 2C 12 4A 50 4A 46   : 02
2550   67 18 08 06 00 0B 66 1C   : 1A
2558   08 06 00 07 66 24 08 06   : AD
2560   00 06 66 18 42 84 60 00   : AA
2568   01 1A 4C DF 07 70 44 FC   : FD
2570   00 00 4E 75 42 80 78 01   : FE
2578   60 00 01 08 78 03 60 00   : 44
-----------------------------------------
SUM:   CB 38 A1 B0 21 07 46 11   50D0

2580   01 02 78 01 60 00 00 FC   : D8
2588   48 E7 0E E0 28 3C 44 22   : E7
2590   44 00 2F 07 2E 3C 3F 80   : A3
2598   00 00 60 26 48 E7 0E E0   : A3
25A0   28 3C 44 28 44 00 2F 07   : 4A
25A8   2E 3C 3F 80 00 00 60 12   : 9B
25B0   48 E7 0E E0 28 3C 44 20   : E5
25B8   44 00 2F 07 2E 3C 40 00   : 24
25C0   00 00 4C F9 07 60 00 00   : AC
25C8   26 54 32 84 4A 50 24 80   : 6E
25D0   4A 50 48 44 32 84 BA 50   : E6
25D8   67 FC 24 87 4A 50 2E 1F   : F5
25E0   60 00 FF 58 48 E7 0E E0   : D4
25E8   28 3C 00 00 44 0E 60 74   : 8A
25F0   48 E7 0E E0 28 3C 00 00   : 81
25F8   44 1D 60 68 48 E7 0E E0   : 46
-----------------------------------------
SUM:   5A 28 2C 85 61 73 2C DA   1079

2600   28 3C 00 00 44 0F 60 5C   : 73
2608   48 E7 0E E0 28 3C 00 00   : 81
2610   44 0A 60 50 48 E7 0E E0   : 1B
2618   28 3C 00 00 44 10 60 44   : 5C
2620   61 00 F6 92 6B 0E 67 14   : DD
2628   48 E7 0E E0 28 3C 00 00   : 81
2630   44 14 60 30 42 80 44 FC   : EA
2638   00 01 4E 75 44 FC 00 05   : 09
2640   4E 75 61 00 F6 70 6B 0E   : 03
2648   67 14 48 E7 0E E0 28 3C   : FC
2650   00 00 44 04 60 0E 42 80   : 78
2658   44 FC 00 01 4E 75 44 FC   : 44
2660   00 00 4E 75 4C F9 07 60   : 6F
2668   00 00 26 54 32 84 4A 50   : CA
2670   24 80 4A 50 32 86 BA 50   : 00
2678   67 FC 20 12 4A 50 60 00   : 8F
-----------------------------------------
SUM:   4D 66 EB 5E BD 2E FD 5B   12E9

2680   FE BA 42 86 32 BC 88 00   : F6
2688   4A 50 24 86 44 C4 4C DF   : 77
2690   07 70 4E 75 00 00 89 00   : C3
2698   A8 00 74 00 00 E9 E0 00   : E5
26A0   00 E9 E0 0A 00 E9 E0 10   : AC
26A8   00 06 00 04 00 12 00 0C   : 28
26B0   00 2E 00 2A 00 1E 00 3E   : B4
26B8   00 16 00 0A 00 0E 00 08   : 36
26C0   00 0C 00 0E 00 0C 00 10   : 36
26C8   00 30 00 52 00 04 00 04   : 8A
26D0   00 04 00 04 00 04 00 04   : 10
26D8   00 04 00 04 00 04 00 04   : 10
26E0   00 04 00 04 00 04 00 04   : 10
26E8   00 04 00 04 00 04 00 04   : 10
26F0   00 04 00 04 00 04 00 04   : 10
26F8   00 04 00 04 00 04 00 04   : 10
-----------------------------------------
SUM:   F7 01 08 3B 76 B8 1D 6D   C6AB

2700   00 04 00 04 00 04 00 04   : 10
2708   00 04 00 04 00 04 00 04   : 10
2710   00 04 00 04 00 04 00 04   : 10
2718   00 04 00 04 00 04 00 04   : 10
2720   00 04 00 04 00 04 00 04   : 10
2728   00 04 00 04 00 04 00 04   : 10
2730   00 04 00 04 00 04 00 04   : 10
2738   00 04 00 04 00 04 00 04   : 10
2740   00 04 00 04 00 04 00 04   : 10
2748   00 04 00 04 00 04 00 04   : 10
2750   00 04 00 04 00 04 00 04   : 10
2758   00 04 00 04 00 04 00 04   : 10
2760   00 04 00 04 00 04 00 04   : 10
2768   00 04 00 04 00 04 00 04   : 10
2770   00 04 00 04 00 04 00 04   : 10
2778   00 04 00 04 00 04 00 04   : 10
-----------------------------------------
SUM:   00 40 00 40 00 40 00 40   A83D

2780   00 04 00 04 00 04 00 04   : 10
2788   00 04 00 04 00 04 00 04   : 10
2790   00 04 00 04 00 04 00 04   : 10
2798   00 04 00 04 00 04 00 04   : 10
27A0   00 04 00 04 00 04 00 04   : 10
27A8   00 04 00 04 00 04 00 04   : 10
27B0   00 04 00 04 00 04 00 04   : 10
27B8   00 04 00 04 00 04 00 04   : 10
27C0   00 04 00 04 00 04 00 04   : 10
27C8   00 04 00 04 00 04 00 04   : 10
27D0   00 04 00 04 00 04 00 04   : 10
27D8   00 04 00 04 00 04 00 04   : 10
27E0   00 04 00 04 00 04 00 04   : 10
27E8   00 04 00 04 00 04 00 04   : 10
27F0   00 04 00 04 00 04 00 04   : 10
27F8   00 04 00 04 00 04 00 04   : 10
-----------------------------------------
SUM:   00 40 00 40 00 40 00 40   A83D

2800   00 04 00 04 00 04 00 04   : 10
2808   00 04 00 04 00 04 00 04   : 10
2810   00 04 00 04 00 04 00 04   : 10
2818   00 04 00 04 00 04 00 04   : 10
2820   00 04 00 04 00 04 00 04   : 10
2828   00 04 00 04 00 04 00 04   : 10
2830   00 04 00 04 00 04 00 04   : 10
2838   00 04 00 04 00 04 00 04   : 10
2840   00 04 00 04 00 04 00 04   : 10
2848   00 04 00 04 00 04 00 04   : 10
2850   00 04 00 04 00 04 00 04   : 10
2858   00 04 00 04 00 04 00 04   : 10
2860   00 04 00 04 00 04 00 04   : 10
2868   00 04 00 04 00 04 00 04   : 10
2870   00 04 00 04 00 04 00 04   : 10
2878   00 04 00 04 00 04 00 04   : 10
-----------------------------------------
SUM:   00 40 00 40 00 40 00 40   A83D

2880   00 04 00 04 00 04 00 04   : 10
2888   00 04 00 04 00 04 00 04   : 10
2890   00 04 00 04 00 04 00 04   : 10
2898   00 04 00 04 00 04 00 04   : 10
28A0   00 04 00 04 00 04 00 04   : 10
28A8   00 04 00 04 00 04 00 04   : 10
28B0   00 04 00 04 00 04 00 04   : 10
28B8   00 04 00 04 00 04 00 04   : 10
28C0   00 04 00 04 00 04 00 04   : 10
28C8   00 04 15 C6 04 66 00 34   : 7D
28D0   00 C2 00 92 00 8E 00 AC   : 8E
28D8   00 AE 00 7C 00 82 00 82   : 2E
28E0   00 A4 00 A2 00 00 00 00   : 46
28E8   00 00 00 00 00 00 00 00   : 00
28F0   00 00 00 00 00 00 00 00   : 00
28F8   00 00 00 00 00 00 00 00   : 00
-----------------------------------------
SUM:   00 3C 15 9A 04 9A 00 86   E411
```

▶最近オクラ入りしていたMZ-80Kを学校に持っていって活用している。学校に持っていくときは疲れました。あれって重いんだもん。　村上　秀広（18）北海道

リスト2 FPPMAC.H

```
  1: _SUPER     equu   $FF20
  2: *
  3: *  define fpp device adress
  4: *
  5: FPPADR     equ   $E9E000
  6: FPPRES     equ   $0
  7: FPPREST    equ   $6
  8: FPPOPREG   equ   $8
  9: FPPCMD     equ   $A
 10: FPPCNDREG equ    $E
 11: FPPDATREG equ    $10
 12: *
 13: * define fpp responce
 14: *
 15: nullres    equ   $0802   * fpp is idle
 16: snddbl     equ   $9608   * request to send double pr. data  to  fpp
 17: sndl4      equ   $9504   * request to send   long integer  to  fpp
 18: recdbl     equ   $b208   *  ready  to send double pr. data from fpp
 19: recl4      equ   $b104   *  ready  to send   long integer  from fpp
 20: *
 21: *
 22: * define  fpp regname
 23: *
 24: fp0      equ     0
 25: fp1      equ     1
 26: fp2      equ     2
 27: fp3      equ     3
 28: fp4      equ     4
 29: fp5      equ     5
 30: fp6      equ     6
 31: fp7      equ     7
 32: *
 33: * define function code
 34: *
 35: __load     equ   $00
 36: __sqrt     equ   $04
 37: __sin      equ   $0E
 38: __cos      equ   $1D
 39: __tan      equ   $0F
 40: __sincos   equ   $30
 41: __arcsin   equ   $0C
 42: __arccos   equ   $1C
 43: __arctan   equ   $0A
 44: __sinh     equ   $02
 45: __cosh     equ   $19
 46: __tanh     equ   $09
 47: __arctanh equ    $0D
 48: __exp      equ   $10
 49: __expm1    equ   $08
 50: __twotox   equ   $11                    * 2^x
 51: __ln       equ   $14
 52: __lnxp1    equ   $06                    * ln( x + 1 )
 53: __log2     equ   $16
 54: __log10    equ   $15
 55: __abs      equ   $18
 56: __neg      equ   $1A
 57: __int      equ   $01
 58: __intg     equ   $03
 59: __getexp   equ   $1E
 60: __getman   equ   $1F
 61: __fadd     equ   $22
 62: __fsub     equ   $28
 63: __fmul     equ   $23
 64: __fdiv     equ   $20
 65: __fmod     equ   $21
 66: __frem     equ   $25
 67: __fscale   equ   $26
 68: __fcmp     equ   $38
 69: __ftst     equ   $3A
 70: *
 71: * define macros
 72: *
 73: *
 74: * wait until required status
 75: *
 76: l_fppadr: macro
 77:         lea      FPPADR,a5
 78:         endm
 79: w_stat: macro    cond
 80:         local    w_loop
 81: w_loop:
 82:         cmp.w    #cond,(a5)
 83:         bne      w_loop
 84:         endm
 85: *
 86: * load double pr. num to fpp with operation
 87: *
 88: lddble: macro    s1,s2,fppreg,op
 89:         w_stat   nullres
 90:         move.w   #$5400+fppreg<<7+op,FPPCMD(a5)
 91:         w_stat   snddbl
 92:         move.l   s1,FPPDATREG(a5)
 93:         move.l   s2,FPPDATREG(a5)
 94:         endm
 95: *
 96: * store fpp to double pr. num to s1,s2
 97: *
 98: stdbl:  macro    s1,s2,fppreg
 99:         w_stat   nullres
100:         move.w   #$7400+fppreg<<7,FPPCMD(a5)
101:         w_stat   recdbl
102:         move.l   FPPDATREG(a5),s1
103:         move.l   FPPDATREG(a5),s2
104:         endm
105: *
106: * move num in sfppreg to dfppreg with operation
107: *
108: movedbl: macro   sfppreg,dfppreg,op
109:         w_stat   nullres
110:         move.w   #sfppreg<<10+dfppreg<<7+op,FPPCMD(a5)
111:         endm
112: *
113: * load long integer to fpp with operation
114: *
115: ldl4:   macro    s1,fppreg,op
116:         w_stat   nullres
117:         move.w   #$4000+op+fppreg<<7,FPPCMD(a5)
118:         w_stat   sndl4
119:         move.l   s1,FPPDATREG(a5)
120:         endm
121: *
122: * store num in fppreg to s1
123: *
124: stl4:   macro    s1,fppreg
125:         w_stat   nullres
126:         move.w   #$6000+fppreg<<7,FPPCMD(a5)
127:         w_stat   recl4
128:         move.l   s1,FPPDATREG(a5)
129:         endm
130: *
131: * load constant to fppreg
132: *     not available ?
133: *
134: ldcnst: macro    fppreg,cnstno
135:         w_stat   nullres
136:         move.w   #$5600+cnstno+fppreg<<7,FPPCMD(a5)
137:         endm
```

リスト3 FRACT.BAS

```
 20 screen 1,2,1,1:console ,,0
 30   float T,U,V,W,X,Y
 50 dim float p(5)={-0.5#,0.5#,-0.5#,0.4#,0.34#,-0.4#}
 60 T=(p(1)-p(0))/511# : U=(p(3)-p(2))/511# : V=p(4) : W=p(5)
 70 for t=1 to 255 : u=cos(pi((t mod 42)/21#))*12+27.5# : if u>31 then u=31
 80    v=cos(pi(((t+21) mod 42)/21#))*7+23.5#  : if v>31 then v=31
 90      palet(t,hsv(190*t/255+1,u,v))
100 /* comment
110    next
180 usefpp( V , W , T , U , p(0) , p(2) , 511 , 255 , &HFF )
```

リスト4 FRFPP.S

```
   1:          include fppmac.h
   2:          .offset 8
   3: recon    ds.l    2
   4: imcon    ds.l    2
   5: dx       ds.l    2
   6: dy       ds.l    2
   7: remin    ds.l    2
   8: immin    ds.l    2
   9: MAXDOT1 ds.l     1
  10: MAXREP  ds.l     1
  11: MASK     ds.l    1
  12:          .XDEF   _usefpp
  13:          .XREF   _pset
  14:          .text
  15: _usefpp:
  16:          link    a6,#0
  17:          clr.l   -(sp)
  18:          dc.w    _SUPER
  19:          move.l  MAXDOT1(a6),d4
  20:          move.l  MAXREP(a6),d5
  21:          move.l  MASK(a6),d6
  22:          move.l  d0,-(sp)
  23:          l_fppadr
  24:          clr.w   FPPREST(a5)
  25:          lddble  recon(a6),recon+4(a6),fp2,__load        * real part of c
onst
  26:          lddble  imcon(a6),imcon+4(a6),fp3,__load        * imaginary part
 of const
  27:          CLR.L   d0
  28: L11:
  29:          CMP.L   d4,d0
  30:          BGT     L12
  31:          CLR.L   d1
  32: L14:
  33:          CMP.L   d4,d1
  34:          BGT     L15
  35:          ldl4    d0,fp0,__load
  36:          lddble  dx(a6),dx+4(a6),fp0,__fmul
  37:          lddble  remin(a6),remin+4(a6),fp0,__fadd
  38:          ldl4    d1,fp1,__load
  39:          lddble  dy(a6),dy+4(a6),fp1,__fmul
  40:          lddble  immin(a6),immin+4(a6),fp1,__fadd
  41:          MOVE.L  #0,d2
  42: L17:
  43:          CMP.L   d5,d2
  44:          BGT     L18
  45:          movedbl fp1,fp6,__load
  46:          movedbl fp6,fp6,__fmul  * Y*Y
  47:          movedbl fp0,fp4,__load
  48:          movedbl fp4,fp4,__fmul  * X*X
  49:          movedbl fp6,fp4,__fsub  * X*X-Y*Y
  50:          movedbl fp2,fp4,__fadd  * Q=X*X-Y*Y+V
  51:          movedbl fp1,fp5,__load
  52:          movedbl fp0,fp5,__fmul  * X*Y
  53:          movedbl fp5,fp5,__fadd  * 2*X*Y
  54:          movedbl fp3,fp5,__fadd  * Q = 2*X*Y + W
  55:          movedbl fp4,fp0,__load
  56:          movedbl fp5,fp1,__load
  57:          movedbl fp4,fp4,__fmul  * Q*Q
  58:          movedbl fp5,fp5,__fmul  * R*R
  59:          movedbl fp4,fp5,__fadd  * Q*Q+R*R
  60:          stdbl   d3,d7,fp5
  61:          cmp.l   #$40100000,d3
  62:          bge     L18
  63: L19:
  64:          ADDQ.L  #1,d2
  65:          BRA     L17
  66: L18:
  67:          and.L   d6,d2
  68: L21:
  69:          movem.l d0-d2,-(SP)
  70:          JSR     _pset
  71:          movem.l (sp)+,d0-d2
  72: L16:
  73:          ADDQ.L  #1,d1
  74:          BRA     L14
  75: L15:
  76: L13:
  77:          ADDQ.L  #1,d0
  78:          BRA     L11
  79: L12:
  80:          unlk    a6
  81:          rts
```

▶先日，愛機X1turboの落下試験を行ってしまいました。それ以来，彼女は機嫌を損ねてなかなか起動してくれません。やはりチェックしてもらうべきでしょうか。なにやら中でカタカタいってるし。

大石　和生（17）静岡県

X1/X1turbo 組曲「くるみ割り人形」よりシナの踊り

Itou Keiichi
伊藤 圭一

X1/X1turbo マリオネット

Sasaki Kouji
佐々木孝司

MZ-2500 ささやきのステップ

Okaue Keisaku
岡上 圭作

さあ，X1用にMIDI対応MMLも発表され，コンピュータミュージックもまた1歩面白い方向へ足を踏み出したようです。今回はX1用に2曲，MZ-2500用に1曲お届けします。なお，X1用は1988年3月号での拡張，MZ-2500用は1987年9月号での拡張が必要です。

まずはクラシック

さて，今月のトップバッターは，以前X1用のMOONLIGHT SERENADEを投稿してくれた伊藤圭一君の作品です。今回はチャイコフスキーの組曲くるみ割り人形から「シナの踊り」に挑戦してくれました。クラシックに興味がないという皆さんも一度くらいは耳にしたことがあるのではないでしょうか。なかなか可愛らしい感じの小曲ですが，どうせならほかの曲にも挑戦してほしいところです(できれば全曲)。

なぜ，シナの踊り（組曲のなかでは，けっこうマイナー）なのかというと，単に曲の作りが単純だったからだそうで……。実際，繰り返し部分が多いためか，非常に短いプログラムにまとまっていますね。

弦楽器はPSG，ベース，バスーン，クラリネット，フルート以外の楽器は無視してあります。フルートのソロは長音に@を多用しているため若干遅くなるところもあり，一部長音が楽譜どおりでないこともありますが，調整用ですので別に間違いではありません(実行には3月号の拡張が必要です)。

リスト中にところどころ見慣れない記号が並んでいます。これはボリュームコントロール用のサブルーチンに渡すコマンドで，アッパーバーは相対ボリュームアップ，アンダーバーは相対ボリュームダウンを意味しています。たとえば，

"C_5"

をA$に入れて，サブルーチン"v"に渡せば，現在のボリューム値よりも5/128だけ小さな音量でCの音を鳴らすMMLが生成されます。MMLのボリュームコントロール用のプリプロセッサだと思ってください。この曲ではアクセントと最後のクレッシェンドにしか使っていませんが，こういった管弦楽曲をMMLに落とすときには有効な手法といえるでしょう。

またもやVIPROOM

続いて，同じくX1(祝版MML要)でBOØWYのマリオネットです。作者の佐々木さんはこのコーナーではお馴染みになったFM音源サークル(?)，VIPROOMの会長という肩書を持っています。以前，渡辺美里の曲で紹介したことがありますね。

なぜ，いまごろ解散してしまったBOØWYなのかというと，たまたま『Keyboard Land』の6月号に楽譜が載ったのを見て，ということらしいのですが，やはり思い入れがないとできないものでしょう。

さて，このプログラムにも変なサブルーチンがあります。"A$ sub"というものですが，これはできあがった曲を友達に聞いてもらったところ，ドラムが弱いといわれ改良した跡だとか。そのほか，マルチプルによる5度の和音などのテクが使われていますが，1音で和音を出す程度はまだFM音源の初級テクニック。やろうと思えば1音でエコーをかけられるようになるというほどFM音源も奥が深いのです(こういうことになるとOh!FMは凄い)。すでに開発されたハイテクをマスターするのもいいでしょうし，新しい技をみつけるのもよいでしょう。誰かキャリアFIXやショートディレイなどにも挑戦してみてください。

最後は歌謡曲

MZ-2500用には薬師丸ひろ子の「ささやきのステップ」です。今回の岡上さんの投稿にはNTTでお馴染みの「あなたをもっと知りたくて」もあったのですが，全体的な完成度でこちらを選びました。クラシック，ポップス，歌謡曲と，我ながら選曲にまったくポリシーがありませんね（博愛主義者と呼んでください)。

演奏にはまず，Oh!MZ 1987年9月号の

薬師丸ひろ子

MML拡張プログラムを使用し，PC-8801シリーズの音色を選択します。

主旋律には2つ重ねたPSGの音をうまく使ってよい音を作っていますね。OPNでのボーカル処理の典型的な例ともいえるでしょう。全体的に素直なプログラムですが，主旋律が奇麗に決まっているので非常に完成度が高く聞こえます。過去に何度かMZ-2500用の投稿には主旋律の使い方，音の割り当て方に問題があるものが多いと指摘しましたが，この作品やドラゴンスピリットではPSGを生かした使い方がされているといえます。ぜひ参考にしてください。

ただし，この作品の場合FM音源の音自体にはまだまだ改善の余地はありそうですね。

お知らせ

さて，著作権がらみで発表できなかった作品名は公開したほうが参考になるのではないかとのお便りをいただきました。なるほど。これまで発表できなかった作品は以下のとおりです。

今回のプログラムのほかに伊藤圭一さんが送ってくれた「アルビノーニのアダージョ」は編曲者のからみで少し難しそうです。そのほか，"シンセの鬼才"マーク・アイシャムの「The Threshold of Liberty」は現

在日本での版権所在不明のため(?)ダメでした。基本的に楽譜が出版されているようなものなら，たいていは大丈夫。マイナーそうな作曲家や20世紀に入ってのクラシックの類は難しいと思っておいてください（日本人なら大丈夫)。

また，今月からMIDIの連載が始まりました。このコーナーでMIDIのデータはどうするのか，などはまだ未定ですがMIDIの投稿も随時受け付けます。楽器名，システム構成を明記し，デモテープを添えて送ってください。

リスト1 シナの踊り

```
10 TEMPO 0:DEFINT a-z
20 DEF FNv$(v)="v"+RIGHT$(STR$(v),LEN(STR$(v))-1)
30 POKE &HAFDE,&HCA:POKE &HAFE1,0:'& - &+
40 'ツギカラ 5ギョウ, オンチョウ ハンプン
50 IF PEEK(&HAE23)<>255 OR PEEK(&HAE24)<>7 THEN 100
60 FOR a=&HAE23 TO &HAE63 STEP 2
70 d=CVI(MEM$(a,2))+1
80 MEM$(a,2)=MKI$(d/2-1)
90 NEXT
100 "s":"0":END
110 '----------
120 LABEL "v"
130 b$=FNv$(v)+a$:j=1+LEN(FNv$(v))
140 FOR i=1 TO LEN(a$)
150 ud$=MID$(a$,i,1)
160 IF ud$="_" OR ud$="¯" THEN v=v+((ud$="_")-(ud$="¯"))*VAL(MID
$(a$,i+1,1)):b$=LEFT$(b$,j-1)+FNv$(v)+RIGHT$(b$,LEN(b$)-j-1):j=j
+LEN(STR$(v))-1:i=i+1
170 j=j+1:NEXT
180 RETURN
190 '----------
200 LABEL "0"
210 PLAY "t63"
220 m1$="r1r2.b-@18>c@18d@19e-@18f@18g@18a@19"
230 m2$="b-4q5a8g8q8gfgfgfgfgfgfgfefg16r8.¯4f4&_4f@26d@25<b-@26f
@25d@26<b-8>r8"
240 m3$="r1r2.>g@21f@21e-@22d@21c@21<b-@22"
250 m4$="a4q5g8a8q8gfgfgfefq5g8a8b-8r8q8<¯4b-4&_4b-@26>d@25f@26b
-@25>d@26f8<r8"
260 s1$="r1r2.rb->rc<rarf>rcrd<rb-rfr4"
270 s2$="r1r2.b-r>cr<arfr>crdr<b-rfrr4"
280 f1$=STRING$(8,"f<b->"):f2$="dddddddddddddd"
290 f3$="e-ce-ce-ce-cdddddddd"
300 b1$="b-rrrb-rrrb-rrrb-rrr"
310 PLAY  "i1o5q8v120L32p1k9"+m1$;
320 PLAY ":i1o5q8v120L32p2k1"+m1$;
330 PLAY "::";
340 PLAY ":i2o3q4v108L8p3k8"+f1$;
350 PLAY ":i2o3q4v108L8p3k2"+f2$;
360 PLAY ":i3o2q4v112L8p1k1"+b1$;
370 PLAY ":i3o2q4v112L8p2k9"+b1$;
380 PLAY ":o5q4v13L8:o4q4v13L8:o3q4v13L8"
390 '
400 a$=m2$+m1$
410 v=120:"v":PLAY b$+":"+b$;
420 PLAY ":::"+f1$+f1$+":"+f2$+f3$+":"+b1$+b1$+":"+b1$+b1$;
430 PLAY ":"+s1$+":"+s1$+":"+s2$
440 '
450 a$=m2$+m3$
460 v=120:"v":PLAY b$+":"+b$;
470 PLAY ":::"+f1$+f1$+":"+f2$+f3$+":"+b1$+b1$+":"+b1$+b1$;
480 PLAY ":"+s1$+":"+s1$+":"+s2$
490 '
500 s3$="r1r2.rfrgrg+ra>rdrc<rb-rfr4"
510 s4$="r1r2.frgrg+rar>drcr<b-rfrr4"
520 a$=m4$+m3$
530 v=120:"v":PLAY b$;:PLAY ":"+b$;
540 PLAY ":::"+f1$+f1$+":"+f3$+f3$+":"+b1$+b1$+":"+b1$+b1$;
550 PLAY ":"+s3$+":"+s3$+":"+s4$
560 '
570 a$=m4$+m1$
580 v=120:"v":PLAY b$;:PLAY ":"+b$;
590 PLAY ":::"+f1$+f1$+":"+f3$+f3$+":"+b1$+b1$+":"+b1$+b1$;
600 PLAY ":"+s3$+":"+s3$+":"+s4$
610 '
620 a$=m2$+m1$
630 v=120:"v":PLAY b$;:PLAY ":"+b$;
640 c1$="b->dfb->d<b-fd<":c2$=">cfa>cfc<af<"
650 c3$="dfb->dfd<b-f":c4$="e-a>ce-ae-c<a"
660 PLAY ":i4o3q8v104L16p3k2"+STRING$(4,c1$)+c2$+c2$+c1$+c1$;
670 PLAY ":i4o3q8v104L16p3k8"+STRING$(4,c3$)+c4$+c4$+c3$+c3$;
680 PLAY ":"+f1$+f1$+":dddddddddddddd"+f3$+":"+b1$+b1$+":"+b1$
+b1$;
690 PLAY ":"+s1$+":"+s1$+":"+s2$
700 '
710 a$=m2$
720 v=120:"v":PLAY     b$+"o5q4v116p3k8 r1r2.f@42&>f@43f@43<";
730            PLAY ":"+b$+"o4q4v116p3k2 r1r2.f@42&>f@43f@43<";
740 PLAY ":"+STRING$(4,c1$)+c2$+c2$+c1$+c1$;
750 PLAY ":"+STRING$(4,c3$)+c4$+c4$+c3$+c3$;
760 PLAY ":"+f1$+f1$+":dddddddddddddd"+f3$+":"+b1$+b1$+":"+b1$
+b1$;
770 s5$=">rc<rarf>rcrd<rb-rf"
780 s6$=">cr<arfr>crdr<b-rfr"
790 PLAY ":r1r2.rb-"+s5$+"rb-";
800 PLAY ":r1r2.rb-"+s5$+"rb-";
810 PLAY ":r1r2.b-r"+s6$+"b-r"
820 '
830 m5$=">f@41&<f@42f@42<f@41&>f@42f@42":m6$=STRING$(2,"f@41&>f@
42f@42<")
840 PLAY     STRING$(4,m5$+m6$);
850 PLAY ":"+STRING$(4,m6$+m5$);
860 PLAY ":"+c2$+c2$+c1$+c2$+c1$+c2$+c1$+c2$;
870 PLAY ":"+c4$+c4$+c3$+c4$+c3$+c4$+c3$+c4$;
880 PLAY ":"+f1$+f1$;
890 PLAY ":e-ce-ce-ce-c"+STRING$(3,"dddde-ce-c");
900 PLAY ":"+b1$+b1$+":"+b1$+b1$;
910 s7$=">rc<rarf>rc<"+STRING$(3,">rd<rb-rf>rc<")
920 PLAY ":"+s7$+":"+s7$+":>cr<arfr>cr<"+STRING$(3,">dr<b-rfr>dr
<")
930 '
940 m7$="¯1>f@41&<f@42f@42¯2<f@41&>f@42f@42"
950 m8$=STRING$(2,"¯1f@41&>f@42f@42<")
960 a$=m7$+m8$+m7$
970 v=120:"v":PLAY     b$;
980 a$=m8$+m7$+m8$
990 v=120:"v":PLAY ":"+b$;
1000 a$=STRING$(3,"¯1b->d¯1fb-¯1>d<b-¯1fd<")
1010 v=104:"v":PLAY ":"+b$;
1020 a$=STRING$(3,"¯1df¯1b->d¯1fd¯1<b-f")
1030 v=104:"v":PLAY ":"+b$;
1040 a$=STRING$(6,"¯1f<¯1b->")
1050 v=108:"v":PLAY ":"+b$;
1060 a$=STRING$(6,"¯1d¯1d")
1070 v=108:"v":PLAY ":"+b$;
1080 a$="t64¯4b-rt65rrt66¯4b-rt67rrt68¯4b-rt69rr"
1090 v=112:"v":PLAY ":"+b$+":"+b$;
1100 a$="¯1>rd<rb->rd¯1<rb->rd<rb-"
1110 v=13:"v":PLAY ":"+b$+":"+b$;
1120 a$="¯1>dr<b-r>dr¯1<b-r>dr<b-r"
1130 v=13:"v":PLAY ":"+b$
1140 '
1150 PLAY "q8v126t60r8>f4:q8v124r8>d4:v116r8f4:v114r8d4";
1160 PLAY ":q8v120r8>f4:q8v120r8>d4:v127r4b-8:v127r4b-8";
1170 PLAY ":o3q8v15r4b-8:o3q8v15r4b-8:o2q8v15r4b-8"
1180 '
1190 END
1200 '----------
1210 LABEL "s"
1220 FOR I=0 TO  7
1230 READ A$:MEM$(&HB190+I*18,18)=HEXCHR$(A$)
1240 NEXT:RETURN
1250 DATA FB00347244311B342B00144E124E0010128C:' Pan FLUTE
1260 DATA 000000000A9A0A28000000000000080000000
1270 DATA EC00704140342511201112141314000008E00:' FAGOT
1280 DATA 0000000008081808000000000C880000200
1290 DATA FC000131000017000F005F5F9F5402820282:' E Bass
1300 DATA 140A0A0A07080708000000000C880000200
1310 DATA F8003132743125252500141414 1D00000000:' SAX
1320 DATA 00000000060606 0A0000000000CC80000200
```

日本音楽著作権協会(出)許諾第8870651-801号

リスト2 マリオネット

```
10 '    ------------------------------------------------
20 '  -------- 「M A R I O N E T T E」  B O O W Y ----------
30 '    ------------------------------------------------
40 IF PEEK(&HAE23)<>255 OR PEEK(&HAE24)<>7 THEN 100
50 '■ オンチョウ ハンプン ■
60 BEEP : FOR X=&HAE23 TO &HAE63 STEP 2
70 MEM$(X,2)=MKI$((CVI(MEM$(X,2))+1)/2-1) : NEXT
80 '■ & <-> &+■
90 POKE &HAFDE,&HCA:POKE &HAFE1,0
100 CLS0 : GOSUB 2690 : R=0
110 TEMPO0 : PLAY"T85R0"
120 GOTO 270
130 '■ MML PLAY ■
140 LABEL "!" : Y$=A$ : "A$ sub"
150 PLAY     A$; : PLAY":"+B$; : PLAY":"+C$; : PLAY":"+D$;
160 PLAY ":"+E$; : PLAY":"+F$; : PLAY":"+G$; : PLAY":"+H$
170 A$=Y$ : RETURN
180 '■ ??? ■
190 LABEL "A$ sub"
200 A=INSTR(A$,"I40") : IF A=0 THEN 230
210 A$=LEFT$(A$,A-1)+"i40v125"+MID$(A$,A+3,LEN(A$)-A)
220 GOTO 200
230 A=INSTR(A$,"I38") : IF A=0 THEN RETURN
240 A$=LEFT$(A$,A-1)+"i38v120"+MID$(A$,A+3,LEN(A$)-A)
250 GOTO 230
260 '■ Intro ■
270 A$="I39P3K3V122O3L16CCDDL8F+F+":B$="I39K8V127O3L16P1CCP2DDP3
L8F+F+":"!"
280 '    "  B.D. :  H.H.  : Bass  : Key  : Key  : Guit. : Guit.
: Vocal"
290 PLAY "I40O1Q5: I36O6Q5: I2O2Q6:   Q7 :   Q7 : I6O4Q7: I6O4Q7
: I3Q8 "
300 PLAY "V120 L8: V120 L8: V115L8:V105  :V110  : V110L8: V110L8
: V113 "
310 A$=STRING$(2,"I40O1C4I38O2CI40O1CR4I38O2C4")
320 B$="O6V125I34C4V120I36"+STRING$(14,"C")
330 C$="O2G+G+B>C+4D+EE+ F+ED+<B4BBB"
340 D$="I11O1L8G+G+RG+&G+2 I12O1F+F+RB&B2
350 E$="I10O2L8D+D+RE&E2 C+C+R<F+&F+2
360 F$="O4L8G+B>D+E4D+@22E@21D+@21C+<B A+F+>F+D+2&D+8"
370 G$="K8"+LEFT$(F$,38)+"O4B2&B8"
380 "!"
390 A$=LEFT$(A$,48)+"I38O2C4C"
400 B$=STRING$(16,"C")
410 F$=LEFT$(F$,41)+"D+"
420 G$=LEFT$(G$,44)+"B"
430 "!"
440 R=R+1:IF R=1 THEN 310
450 A$="I40O1CI38O2CCI40O1C R1I38O2V125CCCV120R"
460 B$="V125I34C8R4C2&C8"
470 C$="O3C+FG+>C+&C+1 V122<F+F+F+V115"
480 D$="R4.O1I12C+&C+2&C+2I11A+A+A+"
490 E$="O2C+FG+>C+&C+2& C+2<C+C+C+"
500 F$="O4C+FG+>C+&C+1 V115<F+F+F+V110"
510 G$="O4C+FG+>K5F&F1 V110<C+C+C+V100"
520 "!"
530 '■ A ■
540 A$=STRING$(2,"L4I40O1CI38O2CI40O1C8C8I38O2C")
550 B$="V125I34C4V120I36"+STRING$(14,"C")
560 C$="O2G+G+G+G+G++G+G+G+>D+D+D+D+D+D+D+D+"
570 D$="I11O1G+2R4Q3G+8G+8Q7 I12F+2R4Q3F+8F+8Q7"
580 E$="I10O2D+2R4Q3D+8D+8Q7 D+2R4Q3D+8D+8Q7"
```

```
590 F$="P1V105L4O4D+RRD+8D+8 D+RRD+8D+8"
600 G$="I6P2V105L4O3G+RRG+8G+8 A+RRA+8A+8"
610 IF R=2 OR R=7 OR R=8 THEN H$="O5L8D+D+D+D+D+D+C+<B >C+4<B>C+
4D+4."
620 IF R=3 THEN H$="O5L8RD+D+EED+C+<B >C+4<B>C+4.D+4"
630 "!"
640 B$=STRING$(16,"C")
650 C$="EEEEEEEE<BBBBBBBB"
660 D$="I11G+2R4Q3G+8G+8Q7 I12B4.B4.B4"
670 E$="E2R4Q3E8E8Q7 <F+4.F+4.F+4"
680 F$="ERRE8E8 D+.D+.D+8E8"
690 G$="BRRB8B8 B.B."
700 H$="RO4BB4BB4G+F+ V108BB4BB4G+"
710 "!"
720 C$="F+F+F+F+F+F+F+F+>C+C+C+C+C+C+C+D+"
730 D$="I11A+2R4A+4 C+4.C+4.C+4"
740 E$="O2F+2R4F+4 G+4.G+4.G+4<"
750 F$="F+F+R8F+8R C+.C+.C+8D+8"
760 G$="O4C+C+R8C+8R <G+.G+."
770 H$="F+4V113F+4G+4A+4 B4A+B4>C+4D+8"
780 "!"
790 IF R=3 OR R=8 THEN 890
800 A$=LEFT$(A$,43)+"8I40O1C8R8C8I38O2C8C8"
810 C$="O2EEEEEEEE F+F+F+F+F+F+F+F+"
820 D$="I12E4.E4.E4 I11A+2A+2"
830 E$="B4.B4.B4 >F+2F+2"
840 F$="EER8<B8>E C+I5F+A+2I6"
850 G$="BBRB F+"
860 H$="EED+D+C+C+<BG+ F+4>F+D+"
870 "!"
880 R=R+1:GOTO 540
890 A$=LEFT$(A$,43)+"8I40O1C8R8C8I38O2C"
900 C$="O2EEEEEEEE F+F+F+F+F+F+F+F+"
910 D$="I12E4.E4.E4 I11A+2I12F+2"
920 E$="B4.B4.B4 >F+2F+2"
930 F$="EER8<B8>E C+2A+2"
940 G$="BBRB F+2>C+2"
950 H$="EED+D+C+C+<BG+ F+4>F+4D+4RD+"
960 IF R=8 THEN H$=LEFT$(H$,26)
970 "!"
980 '■ B ■
990 A$=STRING$(2,"L4I40O1CI38O2CI40O1C8C8I38O2C")
1000 B$="V125I34C4V120I36"+STRING$(14,"C")
1010 C$="O3AAAAAAAA G+G+G+G+G+G+G+G+"
1020 D$="V100I15O1C+1 G+1"
1030 E$="V100I14O1A1 >E1"
1040 F$="P2V110O5C+.AEC+8 <B.>E<BA8"
1050 G$="P1K8"+RIGHT$(F$,24)
1060 IF R=3 THEN H$="O5E4.EEE4E4 C+4<BR4BB&"
1070 IF R=4 THEN H$="O5C+4EE4E4E4 C+4C+4&<BB&"
1080 IF R=8 THEN H$="O5E4.E4E4E4 C+4C+4<BBB&"
1090 IF R=9 THEN H$="O5C+4EE4E4E4 C+4<BR4BB&"
1100 "!"
1110 B$=STRING$(16,"C")
1120 C$="F+F+F+F+F+F+F+F+ EEEEEEEE"
1130 D$="I16D1 I15G+1"
1140 E$="<A1 >E1"
1150 F$="A.>F+D<A8 G+.>E<B." : G$=F$
1160 IF R=3 THEN H$="B&AAA4B&>C+<B4. G+A4B"
1170 IF R=4 OR R=9 THEN H$="BAA4.B&>C+<B4. G+A4B"
1180 IF R=8 THEN H$="B&AAA4B>C+&<B4. G+A4B"
1190 "!"
1200 C$="O3AAAAAAAA G+G+G+G+G+G+G+G+"
1210 D$="I15C+1 I16E1"
1220 E$="<A1 B1"
1230 F$="A>C+8AEC+8 <B8G+>E<BB8" : G$=F$
1240 H$="O5E4F+E4C+4<B4 G+4A4BR>F+"
1250 IF R=4 OR R=9 THEN H$=LEFT$(H$,22)
1260 "!"
1270 IF R=4 OR R=9 THEN 1380
1280 A$="L8I40O1C4I38O2C4I40O1CCI38O2CI40O1C RI40O1CI38O2CI40O1C
RCI38O2C"
1290 B$="CCCCCCCV125I34C4 V120I36CCV125I34C4V120I36CCC"
1300 C$="DDDDDDDD EEEEEEEE"
1310 D$="I16D2.&D8 E2E2&E8"
1320 E$="A2.&A8 B2B2&B8"
1330 F$="P1O5V110A8A.AA8G+2 E8&E2V105"
1340 G$="P2K5O5V110D8D.DD8<B2 B8&B2V100"
1350 H$="O5F+ERF+F+ERE4. R8E4."
1360 "!"
1370 R=R+1:GOTO 990
1380 A$="L8"+STRING$(2,"I40O1CCI38O2CC")+"I40O1CCI38O2CI40O1CC"
1390 B$=STRING$(12,"C")+"I38O2C8O6"
1400 C$="C+C+R4C+C+RC+ V122D+D+D+D+V115R4."
1410 D$="L8I16EER4EER4 I12F+F+F+F+F+"
1420 E$="L8O2C+C+R4C+C+R4D+D+D+D+D+"
1430 F$="P1L8V110C+C+R4C+C+R4V113D+D+D+D+D+"
1440 G$="P2K5L8V110G+G+R4G+G+R4V115A+A+A+A+A+"
1450 H$="O5C+C+R4C+C+R4 V117D+D+D+D+D+"
1460 "!"
1470 '■ C ■
1480 A$=STRING$(2,"I40O1C4I38O2CI40O1CR4I38O2C4")
1490 B$="V125I34C4V120I36"+STRING$(14,"C")
1500 C$="O2G+G+B>C+4D+EE+ F+ED+<B4BBB"
1510 D$="V105I11O1L8G+G+RG+&G+2 A+A+RI12B&B2"
1520 E$="V110I10O2L8D+D+RE&E2 <F+F+RF+&F+2"
1530 F$="O4P1V110L8G+B>D+E4D+@22E@21D+@21C+<B A+B>C+D+&D+2"
1540 G$="O4P2K8V110L8G+B>D+E4D+@22E@21D+@21C+<B A+B>K5F+<B&B2"
1550 H$="O5D+C+D+E4D+C+<B A+A+>F+D+4.R4"
1560 "!"
1570 B$=STRING$(16,"C"):X$=B$
1580 IF R=5 OR R=10 THEN B$=LEFT$(B$,12)+"V125I38O2RCRCV120I36O6
"
1590 D$="I11G+G+RG+&G+2 A+A+RI12V110B4BRB"
1600 E$=">D+D+RE&E2 <F+F+RF+4V115F+RF+"
1610 F$="O4G+B>G+E4D+@22E@21D+@21C+<B A+F+F+F+RV115F+RF+"
1620 G$="O4K8G+B>G+E4D+@22E@21D+@21C+<B A+F+>K5C+<BRV115BRB"
1630 H$="D+C+D+E4D+C+<B A+G+A+BR2"
1640 "!"
1650 B$=X$
1660 D$="V105I11O1L8G+G+RG+&G+2 A+A+RI12B&B2"
1670 E$="V110I10O2L8D+D+RE&E2 <F+F+RF+&F+2"
1680 F$="V110O4G+B>D+E4D+@22E@21D+@21C+<B A+B>C+<B&B2"
1690 G$="V110O4K8G+B>D+E4D+@22E@21D+@21C+<B A+BK5F+F+&F+2"
1700 IF R=10 THEN H$="O5D+C+D+E4D+C+<B A+>F+4D+R4" ELSE H$="O5D+
C+D+E4D+C+<B A+A+>F+D+4.R4"
1710 "!"
1720 IF R=10 THEN 2290
1730 A$=LEFT$(A$,48)+"I38O2V125CRCV120"
1740 B$="CCCCCCCC CCCV125I34CV120I36CCCC"
1750 D$="I11G+G+RG+&G+2 A+A+RI12B&B2"
1760 E$=">D+D+RE&E2 <F+F+RF+&F+2"
1770 F$="O4G+B>D+E4D+@22E@21D+@21C+<B A+F+>C+<B&B2"
1780 G$="O4K8G+B>D+E4D+@22E@21D+@21C+<B A+F+>C+<K5F+&F+2"
1790 IF R=5 THEN 1830
1800 H$="O5D+C+D+E4D+C+<B A+G+A+B4.R4"
1810 "!"
1820 R=R+1:GOTO 1480
1830 C$=LEFT$(C$,24)+"&B2"
1840 H$="O5D+C+D+E4D+C+<B A+G+A+B4.R4"
1850 "!"
1860 A$="I40O1CI38O2CCI40O1CR2 R16I39P1K3Q8V125O3C16C8>C4<R16C16
C8Q5V120"
1870 B$="V125I34C4RC&C2 R4V127C4V120R4"
1880 C$="C+FG+>C+&C+2&C+2.":F$=C$:C$="O3"+C$
1890 D$="R4.I12V115C+&C+2&C+2."
1900 E$="I10O2V115C+FG+G+&G+2& G+2."
1910 F$="O4"+F$
1920 G$="R4.O4G+&G+2&G+2."
1930 H$="R1 R16I39P2K8"+RIGHT$(A$,31)
1940 "!"
1950 '■ D ■
1960 A$=STRING$(2,"L4I40O1CI38O2CI40O1C8C8I38O2C")
1970 B$="V125I34C4V120I36"+STRING$(14,"C")
1980 C$="O2G4BB>CCC+C+ DD<F+F+GGG+G+"
1990 D$="V105"
2000 E$="V110"
2010 IF R=5 THEN F$="P1O3L8G2B>DED4 D>D<D>D4R4":G$="I6K8P2"+RIGH
T$(F$,23)
2020 IF R=6 THEN F$="<D2C+4<B4 A16&B8.AF+A4AB":G$=F$
2030 H$=""
2040 "!"
2050 B$=STRING$(16,"C")
2060 C$="AA>C+C+DDD+D+ EE<G+G+AABG+"
2070 IF R=5 THEN F$="O3A4.A>A4.A G+E<B2RG&":G$=F$
2080 IF R=6 THEN F$=">C+D16C+16<BABAG+F+ E4>F+4E4.F+":G$=F$
2090 "!"
2100 C$="GGGGGGGG >DDDDDDDD"
2110 IF R=5 THEN F$="GGB>DEDF32&F+16.A >D<A>DEF+4.F16F+16":G$=F$
2120 IF R=6 THEN F$="GF+DGF+DGF+ DAF+DAF+D4":G$=F$
2130 "!"
2140 C$="C+C+C+C+C+C+C+C+ <F+F+F+F+F+F+F+F+"
2150 IF R=6 THEN 2190
2160 F$="F1 >D2C+2":G$=F$
2170 "!"
2180 R=R+1:GOTO 1960
2190 A$=LEFT$(A$,29)+"I40O1C8R8R2R8I38O2C8"
2200 B$="CCCCCCCC I38O2CRR2R13406C&"
2210 C$="C+C+C+C+C+C+C+C+ <F+GF+A+>C+<F+GG+"
2220 D$=""
2230 E$="R1 O1F+GF+A+>C+<F+GG+&"
2240 F$="C+C+FFG+G+BB <F+GF+A+>C+<F+GG+&":G$=F$
2250 H$="V113I3Q8"
2260 "!"
2270 R=R+1:GOTO 540
2280 '■ Coda ■
2290 A$="L8I40O1C4I38O2CI40O1CR4I38O2C4 I40O1CRI38O2CI40O1C4I38O
2V125CRCV120"
2300 B$="CCCCCCCC CCCV125I34CV120I36CCCC"
2310 D$="I11G+RG+&G+2 A+A+RI12B&B2"
2320 E$="G+G+R>E&E2 <F+F+RF+&F+2"
2330 F$="O4G+B>D+E4D+@22E@21D+@21C+<B A+F+>C+<B&B2"
2340 F$="O4K8G+B>D+E4D+@22E@21D+@21C+<B A+F+>C+<K5F+&F+2"
2350 H$="D+C+D+E4D+C+<B A+G+A+B4"
2360 "!"
2370 '■ E ■
2380 FOR R=1 TO 4
2390 A$=STRING$(2,"I40O1C4I38O2CI40O1CR4I38O2C4")
2400 B$="V125I34C4V120I36"+STRING$(14,"C")
2410 C$="O2G+G+G+4>C+C+4. <F+F+RB4BB4"
2420 D$="I11G+G+R4G+G+4. A+A+RI12B&B2"
2430 E$="I10O2D+D+R4D+D+4. <F+F+RF+&F+2"
2440 F$="P1O5D+4.D+D+D+4. <F+A+>F+D+&D+2"
2450 G$="I6P2O4F+4.F+G+G+4. K8F+A+>F+<K5B&B2"
2460 H$="I4V110O1BBBB4BB4 R1"
2470 "!"
2480 A$=LEFT$(A$,28)+"I40O1C4I38O2C8I40O1C8R8I38O2V125C8R8C8V120
"
2490 B$=STRING$(16,"C")
2500 C$="G+G+R>C+4C+C+4 <F+F+RB4BB4"
2510 D$="I12BBRI11G+&G+2 A+A+RI12B&B2"
2520 E$="F+F+R>E&E2 <F+F+RF+&F+2"
2530 F$="O4G+G+>D+E4D+@22E@21D+@21C+<B A+B>C+<B&B2"
2540 G$="O4K8G+G+>D+E4D+@22E@21D+@21C+<B A+B>C+<K5F+&F+2"
2550 H$="R1 R2.R8I3G+"
2560 "!"
2570 NEXT R
2580 A$="I40O1CI38O2CCI40O1CR4CI38O2CCI40O1CR4V126CCCR4V127C"
2590 B$="V125I34C4.C4.C4.C4.I38O2V125CCCR4V127C"
2600 C$="O3C+FG+>C+4.<EG+ B>E4.O2V121F+F+F+R4V125G+"
2610 D$="R4.I12O1C+4.R4.>I11G+4.V115A+A+A+R4I10O3V118C+"
2620 E$="O2C+FG+G+4.EG+BB4.V113F+F+F+R4V118G+"
2630 F$="O4C+FG+>C+4.<EG+B>E4.V115F+F+F+R4V120G+"
2640 G$="R4.O4G+4.R4.B4.>V110C+C+C+R4V115D+"
2650 H$=""
2660 "!"
2670 END
2680 '■ TONE SET ■
2690 MEM$(&HB1B4,36)=HEXCHR$("C3 00 65 40 40 30 2F 17 21 00 1F 1
F 1F 1C 0C 00 00 80 00 00 04 06 F6 06 06 06 00 00 00 00 00 00 80
 00 00 00 ") '  2:[ Bass   ]
2700 MEM$(&HB1D8,36)=HEXCHR$("FC 00 72 32 31 72 18 07 1B 00 1D 1
9 1A 1A 04 04 04 84 00 00 00 00 28 27 28 27 00 00 00 00 00 00 80
 00 00 00 ") '  3:[ Vocal  ]
2710 MEM$(&HB1FC,36)=HEXCHR$("C4 00 74 34 35 75 18 00 1B 00 5D 5
9 5A 5D 02 00 08 80 00 00 02 00 16 08 58 07 00 00 00 00 00 00 80
 00 00 00 ") '  4:[ Vocal 2]
2720 MEM$(&HB220,36)=HEXCHR$("FC 00 73 31 31 72 1B 07 0F 00 5D 5
9 5A 5D 02 00 08 80 00 00 02 00 16 08 58 07 00 00 00 00 00 00 80
 00 00 00 ") '  5:[ Guitar ]
2730 MEM$(&HB244,36)=HEXCHR$("D9 00 33 20 44 40 16 18 20 00 5F 5
F 5F 54 80 80 80 80 04 00 00 04 04 04 04 04 00 00 00 00 00 00 80
 00 00 00 ") '  6:[E,Guitar]
2740 MEM$(&HB2D4,36)=HEXCHR$("EC 00 48 48 48 48 07 00 09 7F 9D 9
6 9D 96 08 08 08 08 00 00 00 00 F6 F6 F6 F6 00 00 00 00 00 00 80
 00 00 00 ") ' 10:[   C1   ]
2750 MEM$(&HB2F8,36)=HEXCHR$("EC 00 48 48 46 46 07 00 0C 00 9D 9
6 9D 96 08 08 08 08 00 00 80 80 F6 F6 F6 F6 00 00 00 00 00 00 80
 00 00 00 ") ' 11:[   C2   ]
2760 MEM$(&HB31C,36)=HEXCHR$("EC 00 48 48 4A 4A 07 00 0C 00 9D 9
```

▶いま大学の研究室にMZ-700がある。しかもSUN3とMacIIの間に。ちなみにMW-1を作った森さんも同じ研究室です。　黒木　孝（23）兵庫県

```
6 9D 96 08 08 08 08 00 00 00 00 F6 F6 F6 F7 00 00 00 00 00 00 80
 00 00 00 ") ' 12:[   C3   ]
2770 MEM$(&HB364,36)=HEXCHR$("C4 00 48 48 40 40 16 00 7F 7F 99 9
6 99 96 04 00 04 00 00 00 00 00 24 07 24 07 00 00 00 00 00 00 80
 00 00 00 ") ' 14:[   C+1   ]
2780 MEM$(&HB388,36)=HEXCHR$("C4 00 48 48 46 46 16 00 16 00 99 9
6 99 96 04 00 04 00 00 00 80 80 24 07 24 07 00 00 00 00 00 00 80
 00 00 00 ") ' 15:[   C+2   ]
2790 MEM$(&HB3AC,36)=HEXCHR$("C4 00 48 48 4A 4A 16 00 16 00 99 9
6 99 96 04 00 04 00 00 00 00 00 24 07 24 07 00 00 00 00 00 00 80
 00 00 00 ") ' 16:[   C+3   ]
2800 MEM$(&HB634,36)=HEXCHR$("FB 00 72 0A 0D 31 0D 11 1B 00 1E 1
E 1F 1C 04 02 08 0C C0 00 C0 C0 F2 F1 F4 F6 00 00 00 00 0C C8 80
 00 02 00 ") ' 34:[  Cymb  ]
2810 MEM$(&HB67C,36)=HEXCHR$("FB 00 0E 06 07 00 0F 1B 11 05 1A 1
A 1A 16 04 08 16 92 40 40 80 00 32 72 BA F8 00 00 00 00 00 00 80
 00 00 00 ") ' 36:[  H.H   ]
2820 MEM$(&HB6C4,36)=HEXCHR$("FC 00 5F 55 31 21 00 00 0F 00 9F 5
F 9F 5F 8F 90 92 8F 80 43 43 83 16 D8 C9 A8 00 00 00 00 00 00 80
 00 00 00 ") ' 38:[ Snare  ]
2830 MEM$(&HB6E8,36)=HEXCHR$("F9 00 01 31 32 02 0C 00 02 00 98 9
A 9F DA 0A 10 0C 8B 08 C0 49 00 A7 F6 F5 F5 01 80 00 00 F4 8C 80
 00 02 80 ") ' 39:[ Effect ]
2840 MEM$(&HB70C,36)=HEXCHR$("C0 00 01 0E 00 00 15 2F 07 00 1E 1
E 1E 1D 1A 1C 10 90 40 C0 40 00 FD FE F8 F8 00 00 00 00 00 00 80
 00 00 00 ") ' 40:[ B.Drum ]
2850 RETURN
2860 '  -------------------------------------------------
2870 ' -----  1988 05/29(SUN) 12:34:19    VIP000 CHACHA  -----
2880 '  -------------------------------------------------
```

日本音楽著作権協会(出)許諾第8870651-801号

リスト3 ささやきのステップ

```
10 '
20 '*******************************
30 '*                             *
40 '*    ささやきの  ステップ      *
50 '*                             *
60 '*        薬師丸ひろ子          *
70 '*                             *
80 '*******************************
90 '
100 PLAY INIT:DEF INT A-Z
110 DIM RYTHM1$(9),RYTHM2$(9),A%(4,9)
120 '
130 TEMPO=130
140 TONE LFO 4,1,1,1,3,1
150 TONE LFO 5,1,1,1,-3,1
160 GOSUB 1140
170 '
180 RYTHM1$(0)="t=TEMPO; @7L16o5v9"
190 RYTHM1$(1)="crercerrcrercerr crercerrcrercerr crercerrcrercerr
                crercerrcrercerr"
200 RYTHM1$(2)="errrcrrrcrrrcrrr errrcrrrcrrrcrrr"
210 RYTHM1$(3)="errrcrrrcrrrcrrr L8e.e.ee.e.ee.e.ee4L16"
220 RYTHM1$(4)="crercerrcrercrer"
230 RYTHM1$(5)="crercerrcrercrer crercerrcrercrer"
240 RYTHM1$(6)="crrrrrrrcrrrrrrr crrrrrrrcrrrrrrr crrrrrrrcrrrrrrr
                crrrrrrrcrrrrrrr"
250 RYTHM1$(7)="crrrcrrrcrrrcrrr crrrcrrrcrrrcrrr"
260 RYTHM1$(8)="rrrrcrrrrrrrcrrr rrrrcrrrrrrrcrrr rrrrcrrrrrrrcrrr
            v10 errerrere v9"
270 RYTHM1$(9)="crrrcrrrcrrrcrrr"
280 '
290 RYTHM2$(0)="t=TEMPO; L16y6,15y7,220q3"
300 RYTHM2$(1)="rrerrerrrrerrerr rrerrerrrrerrerr rrerrerrrrerrerr
                rrerrerrrrerrerr"
310 RYTHM2$(2)="errrrrrrrrrrrrrr errrrrrrrrrrrrrr"
320 RYTHM2$(3)="errrrrrrrrrrrrrr L8q1e.e.ee.e.ee.e.ee4q3L16"
330 RYTHM2$(4)="rrerrerrrrerrrer"
340 RYTHM2$(5)="rrerrerrrrerrrer rrerrerrrrerrrer"
350 RYTHM2$(8)="rrrrrrrrrrrrrrrr rrrrrrrrrrrrrrrr rrrrrrrrrrrrrrrr errerrere"
360 '
370 RESTORE 510:GOSUB *MUSIC
380 RESTORE 970:GOSUB *MUSIC
390 RESTORE 570:GOSUB *MUSIC
400 RESTORE 1030:GOSUB *MUSIC
410 RESTORE 730:GOSUB *MUSIC
420 RESTORE 1070:GOSUB *MUSIC
430 END
440
450 *MUSIC
460    READ NO,BASS$,MELODY1$,MELODY2$
470    IF NO=-1 THEN RETURN
480    PLAY RYTHM1$(NO),BASS$,MELODY1$,MELODY2$,MELODY2$,RYTHM2$(NO)
490 GOTO *MUSIC
500 '
510 DATA 0,t=TEMPO; L8o4q8@4v9 ,t=TEMPO; L8o5q8@2v8
520 DATA  t=TEMPO; L16o4q8v14
530 DATA 6,<rb-b-2.&b-1rb-b-2.&b-1,L16b-8b-b-b-8b-b-a8aaa8aag8ggg8gga8aaa8aa
        b-8b-b-b-8b-b-a8aaa8aag8ggg8gga8aaa8aa
540 DATA v12b-2a2g2a2b-2a2g2a2
550 DATA 6,>ra-a-2.&a-1ra-a-2.&a-1,a-8a-a-a-8a-a-g-8g-g-g-8g-g-f8fff8ffg-8g-g-
        g-8g-g-a-8a-a-a-8a-a-g-8g-g-g-8g-g-f8fff8ffg-8g-g-g-8g-g-
560 DATA r4<a-4>a-2.f4e-4d-4{<b->c<b-}4a-2.&a-2.{a-g-f}4
570 DATA 7,f1&f1,f1
580 DATA f1v11f4v10fv9fv8fv7fv6fv5f>o4v14d8v13dv12dv14dv13d
590 DATA 1,r<b-b-2.&b-1rb-b-2.&b-1,L16b-8b-b-b-8b-b-a8aaa8aag8ggg8gga8aaa8aa
        b-8b-b-b-8b-b-a8aaa8aag8ggg8gga8aaa8aa
600 DATA v14f2v13f4v12fv11fv10fv9fv8fv7fv14fv13fv14fv13fv14fv13fv14gv13gv12g
        v11gv14dv13dv14f2v13f4v12f8v11fv10fv9fv8fv7fv6fv14fv13fv14fv13fv14f
        v13fv14gv13gv14fv13fv14fv13fv14dv13d
610 DATA 1,>re-e-2.&e-1< rb-b-2.&b-1,>e-8e-e-e-8e-e-d8ddd8ddc8ccc8ccd8ddd8dd
        <b-8b-b-b-8b-b-a8aaa8aag8ggg8gga8aaa8aa
620 DATA v14e-2v13e-4v12e-v11e-v10e-v9e-v8e-v7e-v14e-v13e-v14e-v13e-v14dv13d
        v14cv13cv14dv13dv14e-v13e-v14cv13cv14d2v13d4v12dv11dv10dv9dv8dv7dv6d
        v5dv4dv3dv2dv1dr8v14d8v13dv12dv14dv13d
630 DATA 1,rb-b-2.&b-1rb-b-2.&b-1,b-8b-b-b-8b-b-a8aaa8aag8ggg8gga8aaa8aab-8b-
        b-b-8b-b-a8aaa8aag8ggg8gga8aaa8aa
640 DATA v14f2v13f4v12fv11fv10fv9fv8fv7fv14fv13fv14fv13fv14fv13fv14gv13gv12g
        v11gv14dv13dv14f2v13f4v12f8v11fv10fv9fv8fv7fv6fv14fv13fv14fv13fv14f
        v13fv14gv13gv14fv13fv14fv13fv14dv13d
650 DATA 1,>re-e-2.&e-1< rb-b-2.&b-1>,>e-8e-e-e-8e-e-d8ddd8ddc8ccc8ccd8ddd8dd
        < b-8b-b-b-8b-b-a8aaa8aag8ggg8gga8aaa8aa
660 DATA v14e-2v13e-4v12e-v11e-v10e-v9e-v8e-v7e-v14e-v13e-v14e-v13e-v14dv13d
        v14cv13cv14dv13dv14e-v13e-v14gv13gv14f2v13f4v12f8v11fv10fv9fv8fv7fv6f
        v5fv4fv3fv2fv1frv14dv13dv14e-v13e-v14fv13f
670 DATA 2,g1g-1,@5v6g1g-1
680 DATA v14gv13gv12gv14gv13gv12gv14cv13cv14g4v13gv12gv11gv10gv14g-v13g-v12g-
        v14g-v13g-v12g-v14dv13dv14g-4v13g-v12g-v11g-v10g-
690 DATA 2,f1e1,f1e1
700 DATA v14fv13fv12fv14fv13fv12fv14fv13fv14b-2v13b-8v12b-v11b-v10b-v9b-v14b-
        v13b-v14b-8v13b-v12b-v14av13av14gv13g
710 DATA 3,f2L8fffff.f.ff.f.ff.f.ff4,f2L8@6v11o5cdeff.f.fg.g.ga.a.ab-4
720 DATA v14g8v13gv12gv11gv10gv14f4.v13fv12fv11fv10fv9fv8fv7fv6fv5fv4fv3fv2f
        v1fr4.r r2.v14dv13dv14e-v13e-
730 DATA 4,<b-.b-.f>c2,@2v9q6o5f.f.cg2
740 DATA v14fv13fv12fv14fv13fv12fv14fv13fv14gv13gv12gv14gv13gv12gv14gv13g
750 DATA 4,d.d.<a>e-2,a.a.e-b-2
```

▶筋肉少女帯は凄い！　こんなパンク聞いたことない。難しいピアノがいい。ハードだ，ストロングだ。SIDE Aの「福耳の子供」は泣ける。“ラッシャー木村はえらい”なんだこの歌詞は？　X68000のCMがやれるのは筋肉少女帯しかいない。畠中　伸之（18）愛知県

```
 760 DATA v14av13av12av14av13av12av14av13av14b-v13b-v12b-v14av13av12av14gv13g
 770 DATA 5,f.f.cf2f.f.cf2,e-.e-.e-e-2e-.e-.e-e-2
 780 DATA v14f8v13fv12fv11fv10fv14e-2v13e-2v12e-v11e-v10e-v9e-v8e-v7e-v14cv13c
          v14dv13d
 790 DATA 4,c.c.<g>d2,e-.e-.<b->f2
 800 DATA v14e-v13e-v12e-v14e-v13e-v12e-v14e-v13e-v14fv13fv12fv14fv13fv12fv14f
          v13f
 810 DATA 4,e-.e-.<b->f2,g.g.da2
 820 DATA v14gv13gv12gv14gv13gv12gv14gv13gv14av13av12av14gv13gv12gv14av13a
 830 DATA 5,<b-.b-.b-a2b-.b-.b->f2,d.d.dd2<b-.>c.df2
 840 DATA v14d2v13d2v12d4v11dv10dv9dv8dv7dv6dv14dv13dv14e-v13e-v14dv13d
 850 DATA 4,c.c.<b->c2,
 860 DATA v14c8v13cv12cv14gv13gv14g8v13gv12gv14f8v13fv12fv14e-v13e-
 870 DATA 4,f.f.cf2,q3de-fb-2
 880 DATA v14f4v13fv12fv11fv10fv9fv8fv14fv13fv14fv13fv14e-v13e-
 890 DATA 4,d.d.<a>d2,
 900 DATA v14d8v13dv12dv14av13av14a8v13av12av14g8v13gv12gv14fv13f
 910 DATA 4,g.g.dg2,de-fg2
 920 DATA v14g4v13g4v12gv11gv14gv13gv14gv13gv14fv13f
 930 DATA 4,e-.e-.<b-> e-.e-.<b->e-1,@0v10o5q8r2e-.f.gb-1
 940 DATA v14e-v13e-v12e-v14fv13fv12fv14dv13dv14e-2v13e-4v12e-v11e-v10e-v9e-
          v14e-v13e-v12e-v14dv13dv12dv14cv13c
 950 DATA -1,,,
 960 '
 970 DATA 6,<rb-b-2.&b-1 rb-b-2.&b-1,@2v8o5L16b-8b-b-b-8b-b-a8aaa8aag8ggg8gga8a
          aa8aab-8b-b-b-8b-b-a8aaa8aag8ggg8gga8aaa8aa
 980 DATA <v14b-2v13b-4v12b-v11b-v10b-v9b-v8b-v7b-v6b-v5b-v4b-v3b-v2b-v1b-r2>
          v12b-2a2g2a2
 990 DATA 6,>ra-a-2.&a-1ra-a-2.&a-1,a-8a-a-a-8a-a-g-8g-g-g-8g-g-f8fff8ffg-8g-g-
          g-8g-g-a-8a-a-a-8a-a-g-8g-g-g-8g-g-f8fff8ffg-8g-g-g-8g-g-
1000 DATA r4<a-4>a-2.f4e-4d-4{<b->c<b-}4a-2.&a-2.{a-g-f}4
1010 DATA -1,,,
1020 '
1030 DATA 8,<b-.b-.b->c2d.d.dc2<b-.b-.b->c.c.cd.d.dc4,@3v10o5q8L8o4b-.b-.b->c2
          d.d.dc2<b-.b-.b->c.c.cd.d.dc4
1040 DATA <v14b-2v13b-4v12b-v11b-v10b-v9b-v8b-v7b-v6b-v5b-v4b-v3b-v2b-v1b-r2
          v12b-8v11b-v12b-8v11b-v12b-v11b->v12c8v11cv12c8v11cv12cv11cv13d8v12d
          v13d8v12dv13dv12dv13c8.v11cv14dv13dv14e-v13e-
1050 DATA -1,,,
1060 '
1070 DATA 1,<rb-b-2.&b-1 rb-b-2.&b-1,@2v8o5L16b-8b-b-b-8b-b-a8aaa8aag8ggg8gga8a
          aa8aab-8b-b-b-8b-b-a8aaa8aag8ggg8gga8aaa8aa
1080 DATA <v14b-2v13b-4v12b-v11b-v10b-v9b-v8b-v7b-v6b-v5b-v4b-v3b-v2b-v1b-r2>
          v12b-2a2g2a2
1090 DATA 1,>ra-a-2.&a-1ra-a-2.&a-1,a-8a-a-a-8a-a-g-8g-g-g-8g-g-f8fff8ffg-8g-g-
          g-8g-g-a-8a-a-a-8a-a-g-8g-g-g-8g-g-f8fff8ffg-8g-g-g-8g-g-
1100 DATA r4<a-4>a-2.f4e-4d-4{<b->c<b-}4a-2.&a-2.{a-g-f}4
1110 DATA 9,f1&f2.fc<b-1&b-1,f1&f1@1o4q8v9d1&d1
1120 DATA f1v11f1v10f1v9fv8fv7fv6fv5fv4fv3fv2fv1f
1130 DATA -1,,,
1140 '
1150 RESTORE 1340
1160 ST=PEEK@(0,$FFF)+1:AD=0
1170 FOR K=0 TO 7
1180   FOR I=0 TO 4
1190     FOR J=0 TO 9
1200       READ A%(I,J)
1210     NEXT J
1220   NEXT I
1230   FOR J=0 TO 9:SWAP A%(2,J),A%(3,J):NEXT
1240   FOR I=1 TO 4:POKE@ ST,AD,A%(I,5):AD=AD+1:NEXT
1250   FOR I=1 TO 4:POKE@ ST,AD,A%(I,7)+(A%(I,8) AND 7)*$10:AD=AD+1:NEXT
1260   FOR I=1 TO 4:POKE@ ST,AD,A%(I,0)+A%(I,6)*$40:AD=AD+1:NEXT
1270   FOR I=1 TO 4:POKE@ ST,AD,A%(I,1)+A%(I,9)*$40:AD=AD+1:NEXT
1280   FOR I=1 TO 4:POKE@ ST,AD,A%(I,2):AD=AD+1:NEXT
1290   FOR I=1 TO 4:POKE@ ST,AD,A%(I,3)+A%(I,4)*$10:AD=AD+1:NEXT
1300   POKE@ ST,AD,A%(0,0),A%(0,2)+A%(0,3)*80,A%(0,4),0,0
1310   AD=AD+5
1320 NEXT K:RETURN
1330 '  0
1340 DATA 58, 15,  2,  1,220,  0,  4,  0,  0,  0
1350 DATA 31,  5,  7,  4,  9, 37,  1,  1, -2,  0
1360 DATA 22,  0,  4,  5,  4, 62,  1,  5,  2,  0
1370 DATA 29,  0,  4,  5,  4, 77,  1,  1,  0,  0
1380 DATA 31,  7,  6,  5,  4,  0,  2,  1,  1,  1
1390 '  1
1400 DATA 28, 15,  2,  0,180,  0,  1,  0,  0,  0
1410 DATA 31, 20,  8, 10,  0, 24,  0,  1,  3,  0
1420 DATA 31, 10,  5, 10,  0,  0,  0,  1,  0,  1
1430 DATA 31, 20,  8, 10,  0, 45,  0,  3,  0,  0
1440 DATA 25, 10,  5, 10,  0,  0,  3,  1,  3,  1
1450 '  2
1460 DATA  2, 15,  2,  0,200,  0,  0,  0,  0,  0
1470 DATA 24, 10,  0,  5, 15, 57,  1, 12,  1,  0
1480 DATA 20, 12,  8,  4,  1, 37,  1,  6,  0,  0
1490 DATA 29, 10,  4,  4,  1, 37,  1,  3, -3,  0
1500 DATA 18, 18,  6,  7,  1,  0,  2,  1,  2,  0
1510 '  3
1520 DATA 58, 15,  2,  0,210,  6,  2,  0,  0,  0
1530 DATA 31, 13,  1,  4, 15, 37,  2,  1,  3,  0
1540 DATA 31, 20,  1, 10, 15, 57,  1, 15,  3,  0
1550 DATA 20, 10,  1,  7, 15, 37,  1,  3,  0,  0
1560 DATA 23,  5,  1,  7, 15,  0,  0,  1,  3,  1
1570 '  4
1580 DATA 32, 15,  2,  0,200,  0,  0,  0,  0,  0
1590 DATA 31,  5,  7,  9,  2, 29,  3,  6, -3,  0
1600 DATA 31,  4,  6,  9,  1, 47,  3,  5, -3,  0
1610 DATA 26,  7,  6,  9,  1, 29,  2,  0, -3,  0
1620 DATA 31,  6,  4,  9,  3,  0,  2,  1, -3,  0
1630 '  5
1640 DATA 24, 15,  2,  0,200,  6,  0,  0,  0,  0
1650 DATA 17, 10, 18, 10,  0, 42,  1, 15,  3,  0
1660 DATA 18,  2,  9, 10,  0, 37,  1,  6,  0,  0
1670 DATA 18,  5,  1,  3,  0, 17,  2,  1,  0,  0
1680 DATA 12,  2,  1,  7,  1,  0,  1,  1,  3,  0
1690 '  6
1700 DATA 58, 15,  2,  0,200,  0,  0,  0,  0,  0
1710 DATA 12,  8,  0, 10,  2, 33,  0,  1,  0,  0
1720 DATA 16, 12,  0, 10,  1, 59,  0,  3,  0,  0
1730 DATA 14, 12,  0, 10,  5, 37,  0,  1,  0,  0
1740 DATA 15, 12,  0,  8,  2,  0,  1,  1,  0,  0
1750 '  7
1760 DATA 44, 15,  0,  0,  0,  0,  0,  0,  0,  0
1770 DATA 30, 25,  0,  5, 15, 10,  0,  0,  0,  0
1780 DATA 30, 15, 15, 10,  5,  0,  0,  0,  0,  0
1790 DATA 30, 25,  0,  5, 15, 10,  0,  0,  0,  0
1800 DATA 30, 15, 15, 10,  5,  0,  0,  0,  0,  0
```

ほいほいファイル術

Iwai Ippei
満開製作所 祝 一平

帰納法によってかどうかは，だーれも知らないけれど，C調言語講座の第2回目です。前回のprintfに続いて，今月は突如としてファイルアクセスの講義だそうです。その名も「ultra.c」。果たしてムーンサルトか満開飛びか，まずは講義を聞いてのお楽しみ。教科書の『K&R』を手元に置くこともお忘れないよーに。

ファイルアクセス

さすがにこのC調言語講座も，第2回目に達したよーである。

さて，普通はこのよーな連載において最初のころはというと，if文やswitch文とかの分岐とか，for文とかwhile文なんぞのループ構造とかをやって，「これこれこーゆー条件のときにこーゆーふーにプログラムの流れが変わるんですよ」というノリで始まりがちである。しかし，私はそんなシケたことはやらないのである。

これはこれで大事なことではあるが，基本的に人間つーもんは面白くないことをやろうとしても，絶対長続きしないものなのである。新皮質では「続けたほうがいい」ということはわかっているのだが，旧皮質のほうが「だって面白くないんだもーん」といってダダをコネてるよーな場合，決して物事は長続きはしないはずなのである(注1)。だから対策は2つしかない。すなわち，「面白くないことは最初からやらない」か，もしくは「なんとかして面白いモノにしてしまう」ことである。

この連載においては，私の長年の研究より開発された「とにかくあと回し法」を採用することとした。よーするにこれは，

1) とりあえず面白いことだけをやる
2) もしも重要なことならば，そのうちやる必要が出てくるであろうから，そのときは“面白くない”などと感じる余裕などはなく，とにかくやる
3) もしもやる必要が出てこなかったならば，そもそも重要なことではなかったということであるから，やらなくてよかった

を3本の柱とした，極めてい一かげんな処世術である。この術式の最大の天敵は「井の中の蛙」という格言であるが，これに対しては「蛙の面に水」で対抗するのである。

注1) ごく稀に，「一度決めたらとことんやり抜く」などという人もいるそーであるが，それは「とにかくやり抜く」ということ自体に快感を感じるという特異体質の人なのであるからして，一般の人（特に小さなお子さん）は危険ですから真似をしてはいけません。そこになんらかの楽しみを発見できなければ，三日坊主→自己嫌悪→諸行無常と3段スライドすることになるでしょう。そう，「なせばなる，何事も」などという非科学的なセリフを信じてはいけないのです（それとも鼻からシイタケヨーグルトを喰えるとでもいうのか）。

驚異の万能プログラム

だからして，第2回目ではいきなりファイルアクセスをやってしまうのである。もしかすると，「そないなことして面白いんかいな」などと関西弁で反論してくる人もいるかもしれないが，やはりこれはこれなりに面白いはずなのである。

さてさて，リスト1に示すのがもっともシンプルであろうというファイルアクセスの例である。これはファイルの内容を表示するプログラムで，ま，いってみれば，typeコマンドみたいなもんだな。しかも使い方によってはcopyコマンドとしても動作するという，タダモノではないプログラムなので，名前もultra.cとした（このよーにすごいプログラムであるから，今月のプログラムのなかでこのリスト1だけはPDSにはしないのである)。

もしかすると読者のなかに，リスト1のどこがファイルアクセスなんだと，いぶかる方もおられるかもしれない。そのよーな場合は急いで，

```
cc  ultra.c[CR]
```

としてコンパイルしたのち，

```
ultra  <  ultra.c[CR]
```

としていただきたい。どーだ，ultra.cの内容が画面に表示されただろう。つーことはtypeコマンドではないか。さらには，

```
ultra  <  ultra.c  >  uc.c[CR]
```

としていただきたい。type コマンドで uc.c を表示してみれば，そこにultra.cが複写されているのを発見するであろう。であるからして，これはcopyコマンドでもあるのだ。う～ん，なんてすごいプログラムであることよ。

あんまりしつこくやるとヒンシュクをかうのでここらでやめておくが，これはよーするにHuman(MS-DOS)におけるリダイレクト機能なのである。Cでは，

getchar( )は，“標準入力”から1文字持ってくる関数
putchar( )は，“標準出力”に1文字出力する関数

なわけだ。でもって，普通は“標準入力”はキーボード，“標準出力”はCRT画面ということになっているのだが，

```
ultra  <  ultra.c[CR]
```

とすると「<」の秘められたパワーが発揮され，“標準入力”がキーボードではなく，ultra.cというファイルに切り換えられるわけだな。さらには，

```
ultra  <  ultra.c  >  uc.c[CR]
```

とすることによって，「>」のパワーも加わり，“標準出力”がCRT画面ではなく（新しく作られた）uc.cというファイルになるわけだ。さてここで（初心者にとって）問題としてお勧めなのが，いきなり，

```
ultra[CR]
```

としてみることである。いろいろキーボードをいじくるとそれなりに学習できるであろう。なお，困ったときに使うとよい魔法を教えておこう。それは，「^Z[CR]」である。これはなかなか便利な呪文であるから，大事に使うように。

ここでリスト1について，2点注意しておかなければならないことがある。まずはプログラムの最初のほうにある。

```
#define  EOF    -1
```

という1行である。これについては，各自で勉強していただきたい。『K&R』だと，15と95ページである(両方読んでおくよーに)。それから，

　(c　=　getchar(　))　!=　EOF

であるが，これは『K&R』の16〜18ページを参考にしていただきたい。ふっふっふっ。

そいでもって，ここで大事なのは，

1)　Cでは，「代入文」自体が「値」を持つ
2)　Cでは演算子の優先順位に気をつけなければいけない

の2点である。つまり，

　int　a, b, c;
　a　=　b　=　c　=　4126;

を実行すると，a，b，cの3変数は，すべて4126という値になるわけだ。そして，『K&R』の18ページにも注意が書いてあるが，カッコを忘れて，

　c　=　getchar(　)　!=　EOF

などと書くと，たちまち思惑とは違う動作をするプログラムとなってしまうのである。それが，リスト2のoka.cである。

これをコンパイルし，

　oka　＜　oka. c[CR]

としても画面には何も表示されないであろう。そこで，

　oka　＜　oka. c　＞　data[CR]

としたあとで，

　dump　data[CR]

とすると，dataのなかには01_Hが詰まっていることを確認できるであろう。これは演算子の優先順位のなせるイタズラなのである。てなわけで，このよーなプログラムをカッコ悪いとゆーのだな。

ところで，“標準入力”，“標準出力”を“ファイルの一種”だと考えれば，やっぱりultra. cはファイルにアクセスしとるわけであるからして，やっぱりファイルアクセスでないわけではないと主張することにやぶさかではないのである。

ultra. cの改良

このように素晴しいultra. cであるが，残念ながら致命的な欠点がある。それは，

　ultra　＜　ultra. x　＞　ultra.q

としたあとで，ultra. xとultra.qのディレクトリを表示させ，ファイルサイズを比較してみるとわかるのである。そう，奇怪なことに，ultra.qのほうがultra. xよりも小さいのである。

このような現象が起きるのは，getchar, putcharが文字ファイル，テキストファイルにアクセスすることを前提として作られているからなのである。だからテキストファイルは問題なくコピーできるのだが，“.x”ファイルなどはうまくコピーできないのである。具体的になにがまずいのかというと，いちばんの問題が$1A_H$を読み込んだ場合の動作なのである。普通のテキストファイルでは，$1A_H$はファイルエンドのマークとして使われるので，こいつに出くわしたとたんに，getcharは勝手にファイルエンドと判断し，−1(=EOF)を返してくるのである。そのよーなわけであるから，途中で終わってしまうのである。

そしてもうひとつ凶悪なのが，$0A_H$と$0D_H$の処理である。getchar(　)は，$0D_H$は無視してなかったことにし，それに対してputchar(　)は$0A_H$を$0D_H$，$0A_H$の2バイトに変換してしまうのである。なんでこーなっているかについては，ビル・ゲイツに聞いてくれである。

そこで，もー少しまともにファイルをコピーするのが，リスト3のutu. cである。こーすれば取りあえずはどんなファイルでもちゃんとコピーできるようになる。使い方は，

　utu　転送元ファイル　転送先ファイル

というふうになっている。つまり，＜と＞はなしということになったわけである。

で，どうやらこのプログラムは，いきなりであるがなかなかに高度である。しかし，これでも，ほとんど実用になる最低限のプログラムであったりする。ではプログラムの説明に入る。まずはいちばん最初の

　#include　＜stdio. h＞

であるが，よーするにこれは，「これから標準入出力ライブラリを使いますよ」という宣言なのである。プログラムのなかほどにある“fopen”などという関数を使うためには，このライブラリが必要なのである。そして，すでに気づいているだろうが，

　#define　EOF　−1

リスト1　ultra. c

```
 1:#define EOF     -1
 2:
 3:main()
 4:{
 5:        int c;
 6:
 7:        while((c = getchar()) != EOF)
 8:                putchar(c);
 9:}
```

リスト2　oka. c

```
 1:#define EOF     -1
 2:
 3:main()
 4:{
 5:        int c;
 6:
 7:        while( c = getchar()  != EOF)
 8:                putchar(c);
 9:}
```

リスト3　utu. c

```
 1:#include          <stdio.h>
 2:
 3:main(argc,argv)
 4:int argc;
 5:char *argv[];
 6:{
 7:        register int c;
 8:        FILE *fp0,*fp1;
 9:
10:        if (argc > 2) {
11:                fp0 = fopen(argv[1],"rb");
12:                fp1 = fopen(argv[2],"wb");
13:                if ((fp0 !=NULL) && (fp1 != NULL)) {
14:                        while((c = getc(fp0)) != EOF)
15:                                putc(c,fp1);
16:                        fclose(fp0);
17:                        fclose(fp1);
18:                        return;
19:                }
20:        }
21:        printf("君は間違えている！¥n");
22:}
```

リスト4　utu. bas

```
 10 str fn0,fn1
 20 int fp0,fp1
 30 char c
 40 input "転送元ファイル";fn0
 50 input "転送先ファイル";fn1
 60 fp0=fopen(fn0,"r")
 70 fp1=fopen(fn1,"c")
 80 if ((fp0<>-1) and (fp1<>-1)) then {
 90   while( not feof(fp0))
100     c = fgetc(fp0)
110     fputc(c,fp1)
120   endwhile
130   fclose(fp0)
140   fclose(fp1)
150 }
160 end
```

の１行がなくなっているのである。どうしてかというと，stdio. h というファイルのなかに，

　＃define　EOF　　(－１)

というのがあるからなのだ（ところで，「stdio」というのは「standard I/O」のハナモゲラである)。

次に，

　main (argc, argv)
　int　argc；
　char　＊argv［ ］；

というのがある。これは何かというと，転送元のファイル名と転送先のファイル名を，プログラム中から参照するために必要な下準備である。詳しくは『K&R』の120ページ以降である（ああ楽ちん)。

次にあるのが，

　register　int　c；

だな。これは『K&R』の89ページである（私の持っている第８刷の索引では99ページの誤植になっているな)。ま，register宣言はなくてもいいものであるから，あまり悩まないよーに。

説明はいよいよ佳境に入るのであるが，相変わらずいいかげんのまま進んでいく。まずは

　FILE　＊fp0，＊fp1；

というのがある。ここでX-BASICを思い出していただきたい。X-BASICでは，fopen関数によってファイルをオープンすると，ファイル番号つーもんが返ってきたわけである。で，そいつを受け取るために，

　int　fp0，fp1

というふうに整数変数の宣言をしとったわけだ。しかし，Cではそれが整数ではなく，「FILEという構造体へのポインタ」になっているのである。

ポインタという呪文を聞いたとたんにめまいを起こした人もいるよーであるが，気を取り直してリスト４のutu. basを見ていただきたい。これはなにかというと，utu. cと同じことをX-BASICでやってみただけなのだ。そしてわかるよーに，fp0，fp1の使われ方は，utu. cとutu. basで，まったく同じなのである。だからいまのところ，悩む必要は全然ないのである。

で，リスト３で説明が必要なのは，fopenの第２引数の“rb”と“wb”の部分であろう。ｒとｗは，読み込むか書き込むのか指定である。

これはX-BASICと似ているから自明であろう。問題はｂのほうであるが，これはどーゆーものかというと「バイナリモードの指定」で，よーするに1AHだろうが0DHだろうが0AHだろうが，よけいな難しいことは考えずにそのまま扱ってしまおうというモードである。

てなわけで，バイナリモードでファイルをオープンすれば，“.x”形式のファイルなどを読み書きしても変なことが起きずにすむというわけである。と，ゆーことは，残る問題はスピードだけつーことになるわけだな。

んでもって，するどい人はわかるよーに，リスト３のプログラムが遅い理由のヘソは，１文字ずつシコシコと「読んでは書く」をしているからなのであった。つまりは，一度に数千字(バイト)ぐらいをどばーっと読んで，そいでもってそれをイッキにどばーっと書けば，速くなるのではないかということが予想されるであろう。

よーするに左の皿から右の皿に豆を移すような作業を考えた場合，一粒ずつ運ぶよりも，何十粒かをまとめてどどっと運んだほうが速いという，極めて当たり前な常識なのである。

では，まずは，どばーっと文字を読み込むにはどーすればいいかである。

そこで出てくるのが，リスト５のhayautu. cである。ここで注目していただきたいのが，fread，fwriteの２関数である。fread はファイルから指定した長さ（バイト数）だけデータを読んできて，配列に格納してくれる関数，fwriteは配列からファイルへ指定した長さだけデータを書き込んでくれる関数である。引数は，

　fread(配列，データサイズ，データ個数，ファイルポインタ)

となっている（fwriteも同じ)。データサイズというのは，「ひとつのデータは何バイトか」ということである。hayautu. cではchar型の配列を使っているので，“１”なわけだ。もしもintかfloatなら４，doubleなら８ということになる。

ちなみにデータサイズはプログラム中でsizeof(char)，sizeof(int)などと書くことも可能で，そのようにするととてもカッコイ

リスト５　hayautu. c

```
 1:#include        <stdio.h>
 2:
 3:#define BSIZE   0x2000
 4:
 5:main(argc,argv)
 6:int argc;
 7:char *argv[];
 8:{
 9:        register int c, sl, dl;
10:        char dbuff[BSIZE];
11:        FILE *fp0,*fp1;
12:
13:        if (argc > 2) {
14:                fp0 = fopen(argv[1],"rb");
15:                fp1 = fopen(argv[2],"wb");
16:                if ((fp0 !=NULL) && (fp1 != NULL)) {
17:                        do{     sl = fread(dbuff,1,BSIZE,fp0);
18:                                if (sl)
19:                                {       dl = fwrite(dbuff,1,sl,fp1);
20:                                        if (sl != dl)
21:                                        {       printf("書き込みエラーです。");
22:                                                return;
23:                                        }
24:                                }
25:                        }while(sl >0);
26:                        fclose(fp0);
27:                        fclose(fp1);
28:                        return;
29:                }
30:        }
31:        printf("君は間違えている！¥n");
32:}
```

イものなのである。

リスト6にそれを使ったプログラムを載せておこう。実行結果は「1，4，4，8，22」となる。そう，sizeof( )には，構造体名を食わせることもできるわけだ。

で，ここまでやってみると，ちょいとまずいことに気がつくであろう。

それは,タイムスタンプである。Human68k(MS-DOS)には，タイムスタンプつーもんがあって，ファイルが作成された（最後に書き換えられた）日付と時間を記録してあるわけだ。で，dir なんぞとするとそのタイムスタンプが表示されるわけである。そして copy コマンドでコピって作ったファイルはタイムスタンプが変わっていないのだが，hayautu. xでコピったファイルだとタイムスタンプがコピった時点での時間に変わってしまっているのである。こーゆーのは気分がよくないので，いきなりここでタイムスタンプの設定へと走るのである。

そして最終的に出来上がったのが，リスト7の maruutu. c である。

hayautu. cから変わったのは，変数宣言の

　int　sfh,　dfh ;

と，終わりのほうで

　fileno( )

　FILEDATE( )

の2つの関数を呼び出していることである。なお，コンパイルするには，/Yオプションを付けて，

　cc　/Y　maruutu. c

としていただきたい。

リスト6　sizemiru. c

```
1: #include        <stdio.h>
2:
3: main()
4: {
5:         printf("%d %d %d %d %d¥n",sizeof(char),
6:                 sizeof(int),sizeof(float),sizeof(double),
7:                 sizeof(FILE));
8: }
```

リスト7　maruutu. c

```
 1: #include        <stdio.h>
 2:
 3: #define BSIZE   0x2000
 4:
 5: main(argc,argv)
 6: int argc;
 7: char *argv[];
 8: {
 9:         register int c, sl, dl;
10:         char dbuff[BSIZE];
11:         FILE *fp0,*fp1;
12:         int     sfh, dfh;
13:
14:         if (argc > 2) {
15:                 fp0 = fopen(argv[1],"rb");
16:                 fp1 = fopen(argv[2],"wb");
17:                 if ((fp0 !=NULL) && (fp1 != NULL)) {
18:                         do{     sl = fread(dbuff,1,BSIZE,fp0);
19:                                 if (sl)
20:                                 {       dl = fwrite(dbuff,1,sl,fp1);
21:                                         if (sl != dl)
22:                                         {       printf("書き込みエラーです。");
23:                                                 return;
24:                                         }
25:                                 }
26:                         }while(sl >0);
27:                         sfh = fileno(fp0);
28:                         dfh = fileno(fp1);
29:                         FILEDATE(dfh,FILEDATE(sfh,0));
30:
31:                         fclose(fp0);
32:                         fclose(fp1);
33:                         return;
34:                 }
35:         }
36:         printf("君は間違えている！¥n");
37: }
```

まずはfileno( )であるが，これはファイルポインタを渡すと，それに対応する“ファイルハンドル”つーもんを返してくれる関数である。タイムスタンプをいじるには，このファイルハンドルが必要なわけだ（のはずである）。

で，次にある，なぜか大文字でできているFILEDATE という関数であるが，これは

　FILEDATE（ファイルハンドル，0）

とすると，そのファイルハンドルのファイルのタイムスタンプを読み出してくる関数である。第2引数の0が露骨である。ここが0でない場合は，タイムスタンプの設定となる。よって，

　FILEDATE (dfh, FILEDATE (sfh, 0))

とすると，sfhのファイルからdfhのファイルへタイムスタンプがコピーされるのである。

Cのライブラリ関数には，これらのファイルアクセスに限らず，さまざまな関数が群雄割拠しているのである。そしてCを究めるには，これらの関数群の制圧と掌握が不可欠であるが，残念ながら未だに私もアフガニスタンだったりする。

今月はそこそこのいーかげんさであった。ではまた来月。

特別講義

Cとアセンブリ言語をリンクして使う

Nakamori AKira
中森 章

先月のC言語特集に引き続いての特別講義，今月はC言語をアセンブリ言語とリンクしてプログラミングするテクニックをご教授願いましょう。状況に応じたアセンブリ言語の活用は，C言語のサイコフレーム（？）としてパワーを発揮するでしょう。

コードを受け入れるだけでは

C言語はアセンブリ言語の代わりに用いられます。しかし，しょせんはコンパイラ言語ですから，最初からアセンブリ言語で書いたプログラムにはスピードの面でもオブジェクト効率の面でもかないません（おや，どこかで見たような書き出しだなあ）。

先月の特集では，Cコンパイラの出力するアセンブリ言語のコードにも学ぶべき点が多いということで，代表的なCの出すコードを調べてみました。が，Cコンパイラの吐き出すコードをそのまま受け入れるだけでは，そこそこの機能を持ったプログラムしか作ることができません。しかし，スピードを要求される部分に対し，アセンブリ言語を隠し味として用いることでプログラム全体の機能アップをすることができるのです。

そこで今回は，一歩進めてC言語をアセンブリ言語とリンクする方法について解説していきたいと思います。

実際，C言語自身もアセンブリ言語とリンクしやすいように作られていますから，そのような使い方もあながち間違った使い方ではないでしょう。C言語はUNIXを記述している言語として有名ですが，実はそのUNIXもスピードを要する部分はアセンブリ言語で記述されているのです。

C言語の常識というメガネで

C言語とアセンブリ言語のリンクは，通常，関数のレベルで行われます。すなわち，C言語で書かれたプログラムに含まれている関数のいくつかをアセンブリ言語で記述した関数に置き換えるという方法です。

関数というものは，それを呼び出す側からは引数（アーギュメント）を与え，関数から戻ってきたときは結果の値を期待します。したがって，呼び出された側（関数）では，引数がどのように渡されてくるのか，そして呼び出した側に対して，結果の値をどのような方法で返したらよいのかを知っていなければなりません。

また，アセンブリ言語で書かれた関数の中からC言語で書かれた関数を呼び出す場合もあるでしょう。嬉しいことに，ほとんどのCコンパイラではこの呼び出し方・呼び出され方の手続きが共通になっています。それはC言語を用いる際の常識といっても差し支えありません。まずは，それらの常識からお話ししましょう。

1）ローカル変数の領域はスタック上に確保される。

はじめに，ここでいうローカル変数とは自動(automatic)変数のことです。関数の入り口で領域が確保され，関数の出口で領域が解放されるので「自動」の名があるのでしょう。関数の先頭で，

```
foo( )
{
        int  i, j;
          :
}
```

のように，staticとかregisterといった記憶クラスの指定をせずに宣言されているiやjが自動変数です（自動変数にするためのautoという宣言もありますが）。

さて，Cコンパイラの出力するコードを眺めると，関数の外部で宣言した外部変数名や関数名は名前の最初か最後に_（アンダースコア）をつけて，

```
_main
```

とか，

```
_foo
```

といったラベルでリスト中に現れています。しかし，自動変数については，プログラムリストのどこにも名前が見当たりません。これは自動変数の領域がスタック上に確保され，自動変数へのアクセスはスタック上のある地点からの相対位置を指定して行われるためです。この自動変数のためのスタック上の領域をスタックフレームと呼び，自動変数へのアクセスの基となる位置をスタックフレームのベースと呼びます。

たとえば，あなたが自動変数として，

```
Chan_Agi_Quess_Paraya
```

などという名前の変数を宣言したとしましょう。しかし，この立派な変数名もCコンパイラのコードのなかでは「スタックフレームのベースから～番目の変数」という情けない名前に変わってしまうのです。スタックフレームのベース（のアドレス）はレジスタに記憶されます。それがフレームポインタと呼ばれるレジスタです。

スタックフレームは関数の入り口で生成され，関数の出口で解放されます。16ビット以上のCPUではこのための専用命令を持っていますが，8ビットや昔の(？)16ビットCPU（お馴染み8086）ではスタックポインタを操作することで同じ処理を実現します。

●スタックフレームを生成する命令

PREPAREとかENTERと呼ばれることが多いのですが，68000ではLINKと呼ばれます。呼び方はいろいろありますが，これらの命令はどれも，

・フレームポインタとして用いるレジスタをスタックに退避（プッシュ）する。
・そのときのスタックポインタの値をフレームポインタにコピーする。
・スタックフレームとして必要なバイト数だけスタックポインタを進める（必要なバイト数を引く）。

という動作を行うという点で一致します。

また，フレームポインタとして使われるレジスタはCPUによってだいたい決まっています。つまり，

8080（Z80）	BC
8086系	BP
68000系	A6

という状況です。それぞれのCPUについてスタックフレーム生成の手順がどのように行われるのかを図1に示しましょう。

ここでは簡略化して自動変数がすべて4バイトのサイズを持っている場合を考えます。このときの各自動変数の位置はフレームポインタからの相対位置で，

1番目の変数　フレームポインタ　－4
2番目の変数　フレームポインタ　－8
3番目の変数　フレームポインタ　－12
4番目の変数　フレームポインタ　－16
　　⋮

となります。

フレームポインタに対してマイナスの変位で参照することに注意しましょう。ただし，Z80（αC）の場合はフレームポインタをスタックフレームを形成した直後のスタックポインタと同じ位置を指していますから変位はプラスとなります（なぜかは知りませんが）。また，フレームポインタ自身は（Z80の場合を除いて）古いフレームポインタが退避されている位置を示しています。

さて，今は変数のサイズが4バイトのもののみを考えたため，フレームポインタからの変位はすべて4の倍数になっています。しかし，1バイトや2バイトなどのいろいろなサイズの変数が一緒に宣言されている場合は，変位の決め方はコンパイラによってまちまちのようです。たとえば，

```
bar( )
{
    int   i;
    char  j;
    int   k;
      ⋮
}
```

という宣言がある場合，iは，

フレームポインタ　－4

の位置に割り当てられ，jは，

フレームポインタ　－8

の位置に割り当てられます（int型を4バイトのサイズと考えた場合）。しかし，次のkは，コンパイラによって，

フレームポインタ　－9

であったり（char型が1バイトであるため，メモリの無駄をしないように割り当てを行うとこうなる），

フレームポインタ　－12

の位置であったりするのです。

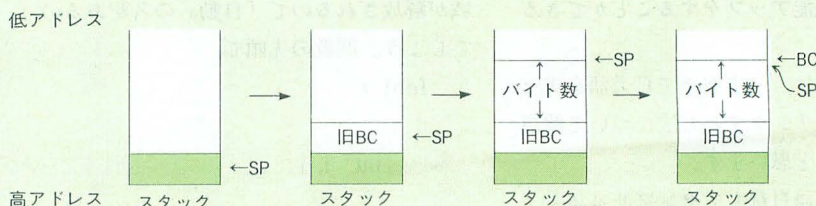
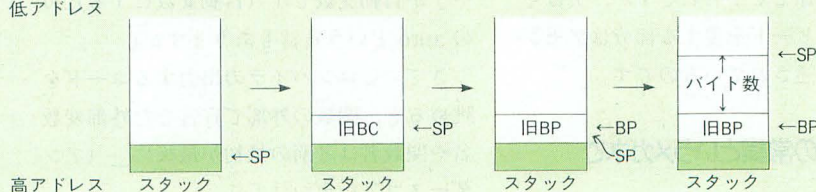
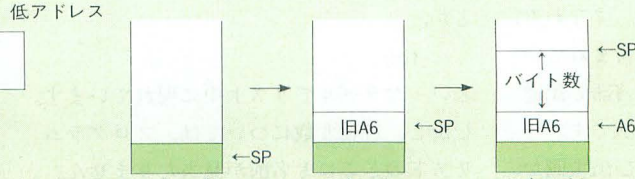
●スタックフレームの解放する命令

DISPOSE，LEAVE，EXITなどと呼ばれます。68000ではUNLKという命令です。スタックフレームの生成は少々ややこしかったのですが，スタックフレームの解放は簡単です。

・フレームポインタをスタックポインタにコピーする。
・スタックから退避しておいた，古いフレームポインタの値を回復（ポップ）する。

この2つの操作でスタックポインタを元の状態に戻すことができます。図2に各CPUのスタックフレームを解放する動作を示します。

ところで，自動変数の知識がアセンブリ言語の関数とC言語のリンクのためになぜ必要なのかわからない人もいると思います。実のところ，自動変数の知識はあまり必要ではありません。まあ，一般教養と思ってください。ただし，フレームポインタとい

図1　スタックフレームの生成

うものが存在するということ，それがスタックポインタを操作して作られるということは知っていなければなりません。なぜなら，このフレームポインタは，スタックフレームのベースへのポインタだけではなく，関数に渡されてくる引数へのポインタとしても用いられるからです。それを次に説明しましょう。

2) 引数はスタックを介して渡される。スタックに引数を積む場合は，int型またはdouble型に変換される。

C言語では引数はスタックを介して関数に渡されます。このときスタックに引数がプッシュされるのですが，引数を積む順序は，関数の引数として後ろのほうに書かれているものが先になります。つまり，

```
foo(a, b, c);
```

という関数呼び出しは，

```
push  c
push  b
push  a
call  foo
```

というコードに展開されます。

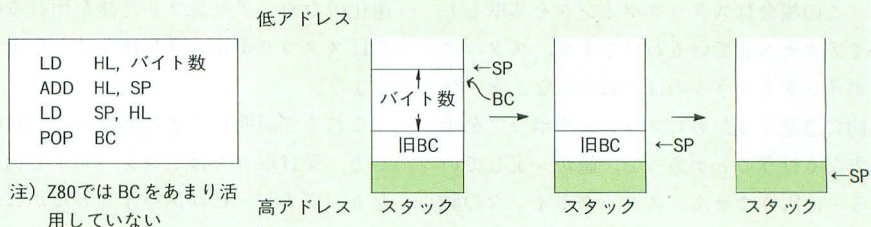
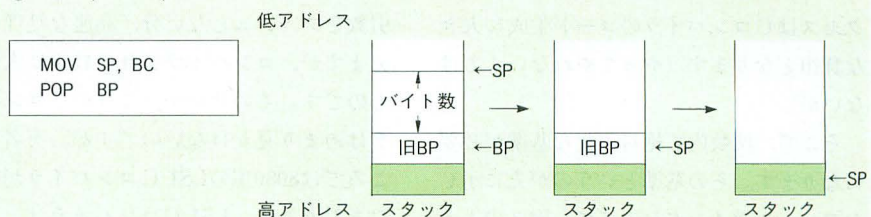
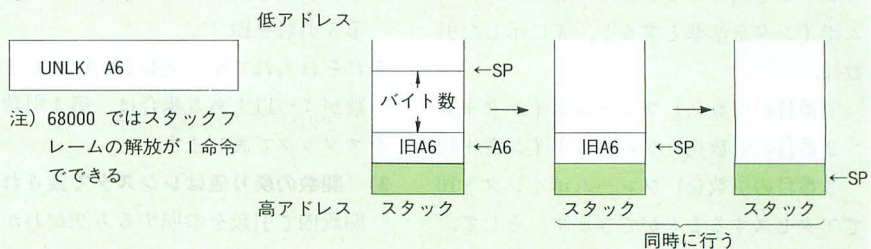
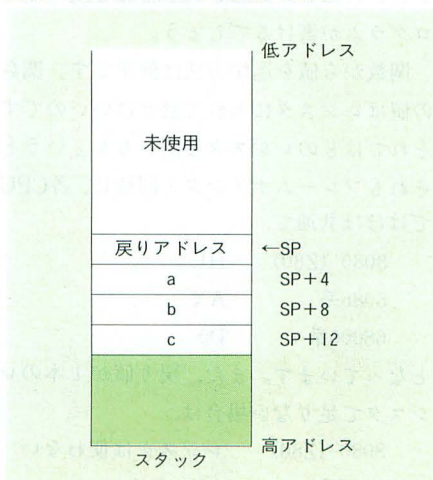
このとき引数は，それが整数型（char, short int, long int）であればlong int型（32ビット）に，それが実数型(float, double）であればdouble型（64ビット）に変換されてスタックに積まれます。

しかし，これはあくまでも原則で（UNIX上のCコンパイラはたいていこうなっていますが），世の中にはびこっているMS-DOS上のCコンパイラは違う方針を取っています。つまり，整数型については，long int型以外はshort int型（16ビット）に変換して積み（long int型はそのまま），実数型についてはfloat型（32ビット）をdouble型に変換することはありません（double型はそのまま）。

これは，UNIX上の多くのCコンパイラが32ビットCPUのためのものであり，MS-DOS上のCコンパイラが16ビットCPUのものであるという点に理由がありそうですが，これがUNIXとMS-DOS間でC言語で書かれたプログラムの移植をする際の大きな問題点となっているのです（UNIXではただ単にintと書いた場合はlong int型を示す）。

また，このことはC言語から呼ばれる関数をアセンブリ言語で書く場合も頭に入れておかなければなりません。つまりスタックに積まれている引数は，UNIXでは4バイト（32ビット）ごとに並び，MS-DOSでは2バイト（16ビット）ごとに並ぶのです（今は整数型の引数のみを考えている）。言い忘れましたが，CP/M-80のCコンパイラはMS-DOSと同じで，Human68kのXCはUNIXと同じです。

さて，先ほどの，

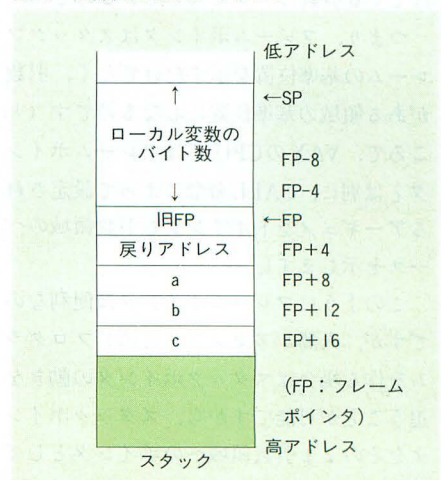
```
foo(a, b, c);
```

という関数呼び出しに戻りましょう。この場合，関数fooの入り口ではスタックの状態は図3のようになっています（引数はすべてlong int型とします)。これにより関数側では，

1番目の引数(a)　スタックポインタ＋4
2番目の引数(b)　スタックポインタ＋8
3番目の引数(c)　スタックポインタ＋12

で引数へのアクセスができるようになります。

図2　スタックフレームの解放

図3　関数の入り口での引数の位置

図4　関数内でのスタックフレーム

リスト1　asm文のサンプル

```
/*
 *      a s m 文のサンプル（このプログラムは実行できません）
 *
 */
main()
{
        int     i,j;

        for(i=0; i<10; i++){
                asm("   move.l   -4(a6),d0  ");
                asm("   add.l    #123,d0    ");
                asm("   move.l   d0,-8(a6)  ");
                printf("i=%d i+123=%d¥n", i, j);
        }
}
```

リスト2　#asm～#endasmのサンプル

```
/*
 *      # a s m ～ # e n d a s m のサンプル
 *      （このプログラムは実行できます）
 *
 */
main()
{
        int     i,j;

        for(i=0; i<10; i++){
#asm
                move.l  -4(a6),d0
                add.l   #123,d0
                move.l  d0,-8(a6)
#endasm
                printf("i=%d i+123=%d¥n", i, j);
        }
}
```

この場合はスタックポインタを基準としてアクセスしているわけですが，スタックポインタというものは，式の値などを一時的に退避するためにプッシュやポップを予告なく行うことがあって，値が一定しているとは限りません。スタックポインタの動きを常に管理していなければならないため，スタックポインタを基準とする引数へのアクセスはCコンパイラのコード生成に大きな負担となります（やってやれないことはないが)。

そこで，関数内で絶対不変な基準が必要になります。その基準というのが先に示したフレームポインタなのです。図3のようなスタックの状態で，スタックフレームを形成すると図4のようになります。フレームポインタを基準とすると，先に示した引数は，

1番目の引数(a)　フレームポインタ＋8

2番目の引数(b)　フレームポインタ＋12

3番目の引数(c)　フレームポインタ＋16

でアクセスすることができます。そして，多くのコンパイラはこの方法で関数に渡されてくる引数にアクセスしているのです。

つまり，フレームポインタはスタックフレームの基準位置を示すだけでなく，引数がある領域の基準位置にもなるのです（ところで，VAXのCPUではフレームポインタとは別に，CALL命令によって設定されるアーギュメントポインタで引数領域のベースを示します)。

このようにフレームポインタは便利なのですが，人間がアセンブリ言語でプログラムを作る場合はスタックポインタの動きを追うことが可能ですから，スタックポインタをそのまま引数領域へのポインタとして利用することもできます。もし，関数の高速化のためにアセンブリ言語を用いるのならばスタックポインタを使うことを考えましょう。

これまで説明してきた関数への引数の渡し方，受け取り方はCコンパイラで標準的なものですが，Cコンパイラのなかにはレジスタを介して関数と引数のやり取りを行うものもあります。引数のレジスタ渡しは引数をプッシュしない分，高速な処理を行えますが，コンパイラ自身の負担は大変なものです。そのせいか，こういうコンパイラはあまり見かけないのですが，有名なところでは8080用のLSI-Cコンパイラがそれにあたります。LSI-Cコンパイラは，

第1引数をHLに，

第2引数をDEに，

第3引数をBCに，

それぞれ入れてから関数をCALLします。引数が4つ以上ある場合は，第4引数以降をスタックで渡します。

3）関数の戻り値はレジスタで渡される。

関数内で引数を参照する方法がわかったら，次は関数から値を返す方法です。これがわかればもうアセンブリ言語で関数のプログラムが書けるでしょう。

関数から値を返す方法は簡単です。関数の値はレジスタに入れて返せばいいのです。それではどのレジスタを用いるかというと，それもフレームポインタと同様に，各CPUではほぼ共通で，

8080（Z80）	HL
8086系	AX
68000系	D0

となっています。また，戻り値が1本のレジスタで足りない場合は，

8080（Z80）	レジスタは使わない
8086系	AX，DX
68000系	D0，D1

となるようです。

これ以外の場合はスタックを使ったりするなどコンパイラごとに工夫が行われていて一般的なことはいえませんが，アセンブリ言語で関数を作る場合には，ほとんどがint型の値を返すと思われるので，これだけ知っておけば十分です。

ちょっとセコいインライン展開

アセンブリ言語を用いて関数を書き，それをC言語のプログラムから呼び出すことは可能です。このとき，どんな小さいプログラムを作る場合でも，最低C言語のプログラムとアセンブリ言語のプログラムの2つのプログラムが必要になります。しかし，多くのCコンパイラではC言語のプログラムにアセンブリ言語の記述を埋め込むことができるようになっています。この機能を使えば，C言語とアセンブリ言語というぐあいにプログラムをわざわざ2つに分けなくても，ひとつのプログラムですんでしまいます。これがアセンブリ言語のインライン展開というやつです。

しかし，このアセンブリ言語の記述はC言語の記述の中に埋め込むだけのものですから，アセンブリ言語だけで書いた場合のような効率のよさは期待できません。また，最適化の段階で用いるオプティマイザが理解できないような命令列を埋め込んでしまう恐れがあるため，インライン展開を用いたプログラムに最適化を行うことはできません。

インライン展開は，新たな関数を作るほどではないが，C言語の文法内ではどうしても無理な機能を実現する場合にのみ使用

すると考えたほうがよいでしょう。アセンブリ言語を使う方法としてはちょっとセコいのですが，何かの役に立つことがあるかもしれませんから説明しておきましょう。

インライン展開の方法はコンパイラによってまちまちですが，大きく分けて次の2つがあります。

1) asm文

UNIX上のCコンパイラはすべて，asm文を使用することでインライン展開をすることができます。asm文とは，

asm("文字列")；

という関数呼び出しのような形式で使用され，( )内に書いてある文字列がそのまま出力コードに埋め込まれます。

パソコンのCコンパイラではasm文によってインライン展開をするものをあまり見かけませんが，LSI-Cは_asm_という名前で始まる関数にasm文と同じ機能を持たせているようです。

asm文の使用例（一応XC用）をリスト1に示しますが，XCはasm文をサポートしていないのでコンパイルするとエラーになってしまいます。UNIX上のCコンパイラで使うときはこのようにすればいいのだという例と思ってください。

なお，asm文で記述してある内容はローカル変数（自動変数）iに123を加えたものを，同じローカル変数のjに代入するという単純なものです。int型の変数はフレームポインタ（A6）からの相対位置で示されることを思い出せば意味はわかりますね。この相対位置は原則として宣言された順に決められるので，リスト1では，

iが－4(A6)に

jが－8(A6)に

なるのです。

2) #asm～#endasm宣言

これはプリプロセッサへの指定のかたちでインライン展開を行う形式です。#asmと#endasmの間の記述がそのまま出力コードに埋め込まれます。パソコンのCコンパイラでインライン展開を行えるものは圧倒的にこの形式が多いようです（XCもこの部類です)。

リスト2にXCによる使用例を示します。これはリスト1でasm文になっている部分をそのまま#asmと#endasmの間に記述したもの（プログラムの意味はまったく同じ）で，こちらはXCでちゃんとコンパイルできますよ。

アセンブリ言語で関数を

それでは，実際にアセンブリ言語で関数を書いてみましょう。例として用いるCコンパイラは，おそらく一番多くの人が持っていると考えられるαCとXCです。ここではフィボナッチ数列を求める関数を変形したものをC言語とアセンブリ言語で記述してその速度を競います。

まずは，リスト3とリスト4にC言語で記述された関数を示します。リスト3がメインプログラムでリスト4の関数を10回呼び出します。そして，私たちはリスト4に対応する関数をアセンブリ言語で記述してリンクすればよいのです。リスト4は再帰呼び出しを行う関数になっていますから，アセンブリ言語で再帰呼び出しを行う場合の参考にもなるでしょう。

1) αC編

αCは直接オブジェクトコードを出力するコンパイラで，そのオブジェクトコードも独自のコードになっていますから，既存のアセンブラで作ったオブジェクトを簡単にリンクすることはできません。

実はαCコンパイラの基になったBDS CコンパイラではアセンブリGOで書かれたソースプログラムをCP/Mに付属のASMにかかる形式に変換するCASM（プリプロセッサ）というユーティリティが付属していました。このユーティリティを用いるとアセンブリ言語のプログラムからBDS Cとリンクできる形式のオブジェクトコードを作ることができたのです。残念ながら，αCにはそのユーティリティは付属していませんが，このコンパイラが最初からアセンブリ言語とのリンクを考えていたものであることがわかるでしょう。

αCはBDS Cとほとんど変わるところがありませんからそのCASMさえ手に入れればアセンブリ言語とのリンクを容易に実現することができます。CASMが手に入らない人は，御手洗毅著『BDS Cプログラミング』(工学図書)を見てください。この本にはCASMをZ80用に改良したZCASMというプリプロセッサの全ソースリストが載っているのでそれを打ち込みましょう。

リスト4の関数をCASMの入力でできるようなアセンブリ言語で記述したプログラムがリスト5です。関数の最初ではスタックポインタ＋2の位置（アドレス）に引数があること（スタックポインタの位置には戻りアドレスがある)，関数の値はHLに入れて返すことに注意すればあとはレジスタの値を壊さない限り何をやってもかまいません。

リスト5では再帰呼び出しのときにスタックに積んだ引数の捨て場所にIXレジスタを使っています。αCは8080用のCコンパイラですから，8080に存在しないレジスタを壊しても問題はないと思います。

このプログラムを，リンクして実行する手順はここでは説明しません。各自マニュアルを調べてください。もっとも，図5に実行結果を示しますが，これを見るだけでも手順はわかると思います。

今，リスト3のプログラム名がFIBMAIN.C，リスト4のプログラム名がFIB1.C，リスト5のプログラム名がFIB2.CSMとなっています。そして，問題の実行時間ですが，fib(20)の値を一度プリントしてから次に再びプリントするまでの時間を比べると，

リスト3　メインプログラム

```
int     fib();
int     i;
main()
{
        for(i=0;i<10;i++)
                printf("fib(20)=%d¥n",fib(20));
}
```

リスト4　メインプログラムから呼ぶ関数(C言語)

```
fib(n)
int         n;
{
            if(n==0) return(0);
            if(n==1) return(1);
            if(n==2) return(1);
            if(n==3) return(2);
            return( fib(n-1)+fib(n-2)+fib(n-3)+fib(n-4));
}
```

C言語のみのときは約28秒であったのが，アセンブリ言語の関数を使うと約10秒になりました。なんと2倍以上のスピードアップです。

ところで，ここで結果として出力されている 16240 という値は正しくありません。これはint型が16ビットしかないために計算の途中でオーバーフローが起きたためです。しかし，間違える条件は同一ですから，さっきの比較も意味があると思われます。

2） XC編

XCの 場 合はアセンブリ言語とのリンクは簡単です。アセンブリ言語で書いたプログラムをHuman68k用のアセンブラ（AS）でアセンブルし，付属のリンカ(LK)で，/Lオプション（リンクの前で止める）をつけてコンパイルしたC言語のメインプログラムとライブラリと一緒にリンクすれば出来上がりです。また，そんな面倒臭いことをしなくてもC言語のプログラムとアセンブリ言語のプログラムを同時に CC に食わせてやればあっという間に実行形式のファイルが出来上がってしまいます。

リスト4のプログラムをアセンブリ言語で記述したものがリスト6です。このプログラムのアルゴリズムはリスト5の8080のものと同じですが，CPU の機能が高い分，すっきりとしたものになっています。

それでは，XC の場合の実行結果を図6に示します。ここではリスト3のプログラム名がfibm. c，リスト4のプログラム名がfib1.c，リスト6のプログラム名がfib2 . sとなっています。ひとつのfib (20) を出力するための実行時間は，C言語だけの場合が約5.3秒で，アセンブリ言語を用いた場合が約3.8秒です。αCのように2倍までと

図5 αCとアセンブリ言語とのリンク(実行例)

```
E>
E>CC FIBMAIN ――― FIBMAIN. Cのコンパイル
BDS Alpha-C Compiler v1.51  (part I)
  32K elbowroom
BDS Alpha-C Compiler v1.51 (part II)
  28K to spare
E>
E>CC FIB1 ――― FIB1. Cのコンパイル
BDS Alpha-C Compiler v1.51  (part I)
  32K unused
BDS Alpha-C Compiler v1.51 (part II)
  28K to spare
E>
E>CLINK FIBMAIN FIB1 ――― リンク
BDS Alpha-C Linker  v1.51
Linkage complete
  39K left over
E>
E>FIBMAIN ――― 実行
fib(20)=16240 ┐
fib(20)=16240 ┘この間約28秒
fib(20)=16240
fib(20)=16240
fib(20)=16240
fib(20)=16240
fib(20)=16240
fib(20)=16240
fib(20)=16240
fib(20)=16240

E>CASM FIB2 ――― FIB2. CSMをASMにかかる形式に変換
BD Software CRL-format ASM Preprocessor v1.50
Processing the FIB function...        FIB2.ASM is ready to be assembled

E>ASM FIB2 ――― FIB2. ASMをアセンブル
CP/M ASSEMBLER - VER 2.0
S0003 =         SECTORS$ EQU ($-TPALOC)/256+1 ;USE FOR "SAVE" !.
0108
001H USE FACTOR
END OF ASSEMBLY

E>MRID FIB2.HEX ――― FIB2. HEXをメモリにロード(MRIDを使っているが
Armat Z80 Debugger - VER 1.0                       DDTでよい)
  (c) Armat co. 1986 All Rights Reserved.

 Reading: FIB2.HEX  Next  = 0108
-G0           ――― MRIDを抜ける (DDTでも同じ)

E>SAVE 3 FIB2.CRL ――― FIB2. CRLを作る
E>
E>CLINK FIBMAIN FIB2 ――― リンク
BDS Alpha-C Linker  v1.51
Linkage complete
  39K left over
E>
E>FIBMAIN ――― 実行
fib(20)=16240 ┐
fib(20)=16240 ┘この間約10秒
fib(20)=16240
fib(20)=16240
fib(20)=16240
fib(20)=16240
fib(20)=16240
fib(20)=16240
fib(20)=16240
fib(20)=16240

E>
```

図6 XCとアセンブリ言語とのリンク(実行例)

```
A>
A>cc /O fibm.c fib1.c
X68k CC Driver v1.00 Copyright 1987 SHARP/Hudson
fibm.c
X68k CCP Pre-Processor v1.00 Copyright 1987 SHARP/Hudson
X68k CC0 Parser v1.00 Copyright 1987 SHARP/Hudson
fibm.c  7       :Warning 15 関数の戻り値がない
X68k CC1 Code Generator v1.00 Copyright 1987 SHARP/Hudson
X68k CC2 Optimizer v1.00 Copyright 1987 SHARP/Hudson
X68k Assembler v1.00 Copyright 1987 SHARP/Hudson
No Fatal error(s)
fib1.c
X68k CCP Pre-Processor v1.00 Copyright 1987 SHARP/Hudson
X68k CC0 Parser v1.00 Copyright 1987 SHARP/Hudson
X68k CC1 Code Generator v1.00 Copyright 1987 SHARP/Hudson
X68k CC2 Optimizer v1.00 Copyright 1987 SHARP/Hudson
X68k Assembler v1.00 Copyright 1987 SHARP/Hudson
No Fatal error(s)
X68k Linker v1.00 Copyright 1987 SHARP/Hudson
A>
A>fibm
fib(20)=147312 ┐
fib(20)=147312 ┘この間約5.3秒
fib(20)=147312
fib(20)=147312
fib(20)=147312
fib(20)=147312
fib(20)=147312
fib(20)=147312
fib(20)=147312
fib(20)=147312
A>
A>
A>cc /O fibm.c fib2.s
X68k CC Driver v1.00 Copyright 1987 SHARP/Hudson
fibm.c
X68k CCP Pre-Processor v1.00 Copyright 1987 SHARP/Hudson
X68k CC0 Parser v1.00 Copyright 1987 SHARP/Hudson
fibm.c  7       :Warning 15 関数の戻り値がない
X68k CC1 Code Generator v1.00 Copyright 1987 SHARP/Hudson
X68k CC2 Optimizer v1.00 Copyright 1987 SHARP/Hudson
X68k Assembler v1.00 Copyright 1987 SHARP/Hudson
No Fatal error(s)
fib2.s
X68k Assembler v1.00 Copyright 1987 SHARP/Hudson
No Fatal error(s)
X68k Linker v1.00 Copyright 1987 SHARP/Hudson
A>
A>fibm
fib(20)=147312 ┐
fib(20)=147312 ┘この間約3.8秒
fib(20)=147312
fib(20)=147312
fib(20)=147312
fib(20)=147312
fib(20)=147312
fib(20)=147312
fib(20)=147312
fib(20)=147312
A>
A>
A>
```

はいきませんがかなりのスピードアップです。

アセンブリ言語は最後の武器だ?

アセンブリ言語とC言語とのリンクを行えば，プログラムの実行速度を上げられることがわかったと思います。それならば，最初からアセンブリ言語だけでプログラムを作るほうがいいという考えも当然出てきますが，アセンブリ言語だけでC言語に匹敵するような制御構造を実現しようとすると大変な労力が必要となります。それに，まったくスピードを要求されない部分までもアセンブリ言語で記述するのは労力の無駄とも言えます。

やはり，アセンブリ言語はどうしてもスピードが要求されるプログラムのクリティカルパスに対して用いたとき（当然，プログラムはよくモジュール化されていて，アセンブリ言語で書くべき関数が明確になっていなければなりませんが）に，一番の能力を発揮するものなのでしょう。

ところで，この原稿ではC言語とアセンブリ言語のリンクがさも当たり前に行われているように書いてきました。しかし，現実にはよほどのことがない限りC言語の出力するコードで満足するプログラマが多いようです。

たとえば，VAXのUNIX上でC言語を使用している人のうち，どのくらいの人がそのCPUのアセンブリ言語の命令を知っているのでしょう。少なくとも私の周りにはVAX上でアセンブリ言語を活用している人は皆無です。きっと，「C言語はアセンブリ言語の代用に使っている」なんていう人のなかにも，アセンブリ言語（そのCPUはなんにせよ）を知らない人が多いのではないでしょうか（これは一種の詭弁ではないだろうか）。

でも，Cの性能によってはどうしても高速化したいという人もいるかもしれませんが，もしかしたらアセンブリ言語とC言語のリンクというテクニックはかなり「その筋」なことなのかもしれませんね。

リスト5　メインプログラムから呼ぶ関数（αC, CASM用）

```
        FUNCTION        fib
fib:
        push            d         ; DE : register variable
        lxi             h,4
        dad             sp
        mov             a,m
        inx             h
        mov             h,m
        mov             l,a
        ora             h
        jz              L0
        dcx             h
        push            h         ; (SP) = HL-1
        mov             a,l
        ora             h
        jz              L1
        dcx             h
        push            h         ; (SP) = HL-2
        mov             a,l
        ora             h
        jz              L2
        dcx             h
        push            h         ; (SP) = HL-3
        mov             a,l
        ora             a
        jz              L3
        dcx             h
        push            h         ; (SP) = HL-4
        call            fib
        xchg
        db              0ddh,0e1h ; POP IX
        call            fib
        dad             d
        xchg
        db              0ddh,0e1h ; POP IX
        call            fib
        dad             d
        xchg
        db              0ddh,0e1h ; POP IX
        call            fib
        dad             d
        db              0ddh,0e1h ; POP IX
        pop             d
        ret
L0:
        pop             d
        ret
L1:
        pop             h
        lxi             h,1
        pop             d
        ret
L2:
        pop             h
        pop             h
        lxi             h,1
        pop             d
        ret
L3:
        pop             h
        pop             h
        pop             h
        lxi             h,2
        pop             d
        ret
        ENDFUNC
```

リスト6　メインプログラムから呼ぶ関数（X68000アセンブリ言語）

```
        .globl   _fib
        .text
        .even
_fib:
        move.l   4(sp),d0
        lea      -20(sp),sp
        move.l   d7,16(sp)
        cmpi.l   #0,d0
        beq      L0
        subq.l   #1,d0
        beq      L1
        move.l   d0,12(sp)          ; n-1
        subq.l   #1,d0
        beq      L2
        move.l   d0,8(sp)           ; n-2
        subq.l   #1,d0
        beq      L3
        move.l   d0,4(sp)           ; n-3
        subq.l   #1,d0
        move.l   d0,(sp)            ; n-4
        jsr      _fib
        move.l   d0,d7
        addq.l   #4,sp              ; +4
        jsr      _fib
        add.l    d0,d7
        addq.l   #4,sp              ; +4 (8)
        jsr      _fib
        add.l    d0,d7
        addq.l   #4,sp              ; +4 (12)
        jsr      _fib
        add.l    d7,d0
        lea      8(sp),sp           ; +8 (20)
        move.l   -4(sp),d7
        rts
        .even
L0:
        lea      20(sp),sp          ; d7 no change
        clr.l    d0
        rts
        .even
L1:
        lea      20(sp),sp          ; d7 no change
        moveq.l  #1,d0
        rts
        .even
L2:
        lea      20(sp),sp          ; d7 no change
        moveq.l  #1,d0
        rts
        .even
L3:
        lea      20(sp),sp          ; d7 no change
        moveq.l  #2,d0
        rts

        .end
```

X68000

今回はX68000用のMS-DOSエミュレータCONCERTOと2つのデータベースを紹介しましょう。なお,このコーナーは今月で終了です。来月からX68000関係の情報はSOFTOUCHや一般記事としてお伝えします。長らくご声援ありがとうございました。

CONCERTO-X68K
DATA PRO-68K/CARD PRO-68K

X68000あなたの知らない世界

MS-DOSエミュレータ CONCERTO-X68K

ついに,アクセスからMS-DOSエミュレータCONCERTO-X68Kが発売になりました。これはX68000上でMS-DOSのソフトを走らせてしまおうというものです。MS-DOSのソフトが動いてなにがおいしいのだろうと思う人もいれば,エミュレータなどは遅くてしょうがないだろうという人もいるでしょう。はたして,そうなのか。今まで,何度かこのCONCERTO-X68Kについてのレポートが本誌でされていました。今回,本格的に試用する機会がありましたので,詳しく探ってみたいと思います。

まずはCONCERTO-X68Kの構成

CONCERTO-X68Kの薄緑色のパッケージを開けると,中にはDOS Engineと呼ばれるLSI満載のボード,ディスケット(マスターディスク)1枚,マニュアル,その他アンケートハガキなどが入っていました。要するに,このCONCERTOはハード,ソフト表裏一体の代物なのです。それでは,まずDOS Engineのほうから見ていきたいと思います。

DOS Engineは30cm×17cmのかなり大きなボードで,X68000の後方にあるスロットにはとうてい収まらない大きさです。ではこのボードはどうしてこんなに大きいのかというと,このボードにはマイクロプロセッサV30と512Kバイト分のメモリなど,通常はコンピュータの心臓部となるべきものが詰まっているからです。はっきりいってこのCONCERTOはMS-DOSエミュレータというよりは,MS-DOSマシンの心臓部だけをみつくろってボードに載せたようなものです。モニタとキーボードのないコンピュータと考えたほうがいいでしょう。こう考えると,約10万円という価格もうなずけます。でもこれを使うには別にMS-DOSのソフトが必要なので,ちょっと高いような気がしますが。

CONCERTOに搭載されているCPU,V30は8086との完全互換性のあるNEC製の16ビットマイクロプロセッサです。このボードではV30はクロック周波数8MHzで駆動されています。ということはPC-9801シリーズのベストセラーモデルであるVM2など(10MHz)に比べれば,若干遅くなっています。X68000本体が10MHzですので,少なくともそれにあわせてほしかったところですが,クロックを上げるとそれにあわせてメモリの値段が上がってしまうので価格の兼ねあいからしかたがないのかもしれません。メモリは512Kバイトを装備していますが,実際にアプリケーションソフトの実行に使えるのは480Kバイト程度です。あとはシステムなどが使っているのでしょう。

次はソフトのほうを見てみたいと思います。マスターディスクには,このDOS Engineを操作するために,

XDOSINIT.X
XDOS.X
XDOS.CNF

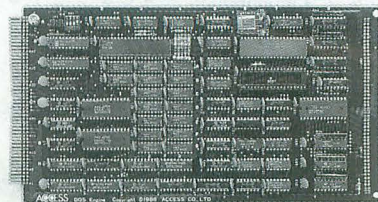
というファイルがついてきます。これだけでDOS Engineは十分に操作できます。なかなかよくできていますね。では,これらについて説明してみましょう。

XDOSINIT.XはDOS Engineをエミュレータとして初期設定をするためのソフトです。これはCONCERTOを立ち上げるたびに実行する必要があります。

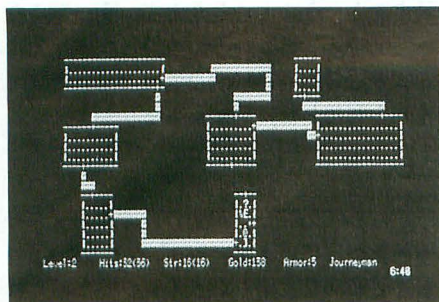
XDOS.Xは実際にDOS Engineを使ってMS-DOSのエミュレーションを行うためのものです。これはMS-DOSのソフトを実行するたびに使うことになります。

XDOS.CNFはXDOS.Xを起動するときに,その環境を設定するためのパラメータ(環境変数など)を設定するためのファイルです。なくてもかまいませんが,その場合はデフォルト値が設定されることになっているようです。

なにが走るのであろうか

MS-DOSのアプリケーションソフトといっても,いろいろなものがあります。また,PC-9801のソフトウェアのなかには,単にMS-DOSをソフトの立ち上げのためだけに使っているものがたいへん多いので気をつけなければなりません。注意してください。あらかじめことわっておきますが,一太郎などはPC-9801のアプリケーションであって,MS-DOSのアプリケーションとはいいがたいものがあります。当然,こういった特定機種用のプログラムはCONCERTOでは実行できません。

これがDOS Engine

このCONCERTOではMS-DOSのVer2.11上で走るアプリケーションソフトが実行可能です。現在広く使用されているMS-DOSはVer3.1程度のものですが,実際にはMS-DOSのソフトの多くはVer2.1の時代に作成されたものです。また一般にVer3.1は広い記憶容量を必要とし,アプリケーションソフトを満足に走らせることができないことがあります。その点からいうと,Ver2.11というのは賢明な選択であると私は思います。しかし,バージョンの違いはCONCERTO上で動くアプリケーションソフトの決定に重要な影響を与えますので十分に注意してください。

では具体的にどのようなソフトが動くのかということに話を進めましょう。MS-DOSのファンクションコールでは基本的にキャラクタしか表示できませんので,グラフィックを使うものは特定のハード用のソフトというのが正しいのです。CONCERTOはあくまでもMS-DOSエミュレータですから,このようなソフトには対応できるわけ

ROGUEが走る!

がありません。

次にMS-FORTRANやMS-PASCALに代表されるコンパイラやインタプリタなどのソフトですが，これらはほとんど問題なく動きます。ただ注意しなければならないことが2つあります。

ひとつはTURBO PASCALなどのようにエディタを内蔵したものや，インタプリタなどのカーソルエディット機能を持っているようなものです。これらは画面やカーソルを制御する方法をインストールできるものか，たまたまX68000と同じものを使っている場合ならば問題なく動きます（ただしインストールは必要でしょう)。また画面制御にANSI.SYSというドライバを用いているものがありますが，これは画面制御のエスケープシーケンスコードをANSI規格のもので制御できるようにするものです。でもX68000は画面制御にANSI規格のものを用いていますので，このANSI.SYSドライバは不要です。それから，ついでにいっておくと，MS-DOSのドライバの接続が必要なソフトは走らないか，走ってもそのドライバを使う機能は使えなくなります。これはCONCERTOがあくまでもエミュレータなのでしかたがありません。

それからワープロなどのときと同様に最初から特定のハードで動くように作られているものは，やっぱりだめです（TURBO Cなどがそのようです)。ちなみにMBASICは由緒正しい昔のカーソルエディットのできないBASIC（昔はみんなそうだった）ですので，ちゃんと走ります。

さて，もうひとつはコンパイラの使用するライブラリ，特に8087などを使ったライブラリです。これらは通常割り込みを用いて処理を行います。このCONCERTOは割り込みのレベルなども変更できるような設計になっているのですが基本的にはIBM-PCなどのソフトで使用されているNMI割り込みを使うものと，割り込みを使わないものが使用可能です。したがって，MS-FORTRANやMS Cのライブラリは IBM-PC用のものを選択し(セッティングし)，使用するようにしてください。

なお，発売元のアクセスからは以下のようなソフトが動くことが明示されています。

MS C　MS-FORTRAN
MS-PASCAL　MS-LINK　MASM
Lattice C　R/M FORTRAN
PLINK86　TURBO PASCAL

さて次にエディタについては，どうでしょうか。はっきりいってちまたにあふれているエディタは機種に依存したものが多く，あまり期待しないほうがよさそうです。おそらく，MS-DOSでなにもせずに使えるエディタとしては，かの悪名高きラインエディタのEDLINぐらいしかないと思われるからです（私はそんなに使いづらいとは思わないんだけどなぁー)。実際には，先ほどのTURBO PASCALなどのところで述べましたように，画面の制御方法が問題となります。特にX68000の場合には，画面が96文字×32行となっていますので（ふつうは80文字×25行)，これがインストールできるかという問題があります。こういうふうに考えていくといちばん安全なのがTURBO PASCALのエディタのような気がします。もちろん，microEMACSを使って，Humanと操作環境を同じにするというのも非常に建設的な意見です。

最後にROGUE，HACKなどに代表されるキャラクタグラフィックのゲームですが，これらは若干のコンフィギュレーションファイルの書き換えを要するものの，だいたい動くようです。ただ，ROGUEはPC-9801版とMS-DOS版の2つがあるようなので注意してください。

いったい使い心地はどうなのか

では，実際にCONCERTOを使ってみたいと思います。まずはボードの取り付けですが，これは簡単です。X68000の電源を落とし，本体後方のスロットのふたをはずし，ボードをぐっと差し込んでネジを締めるだけ。またマニュアル（取り扱い説明書）もそれなりに親切です（あんまり説明することがないからなぁ)。

では試しにマスターディスクから立ち上げてみましょう。すると，すかさずAUTOEXEC. BATでXDOSINIT.Xが実行されてそのことのメッセージが画面に出力されます。そしてプロンプトが出てCOMMAND.Xの入力待ちとなります。ここで，おもむろにBドライブのTURBO PASCALを実行したいときには，

XDOS B：TURBO

と入力します。するとマスターディスク内のXDOS.Xが立ち上がって，TURBO PASCALが実行されます。

それではシステムディスクに組み込んでみましょう。ファイルの移動を開始しました。XDOSINIT. Xは基本的にシステムが立ち上がるときに一度実行されればいいので，ディレクトリSYSへ移動。XDOS.Xは繁頻に使うのでディレクトリBINにXDOS.CNFとともに移動しました。

次にCONFIG. SYSを書き直します。というのはこのXDOS.Xが作動中はXDOS.X自体がファイルを開き，XDOS.Xで動作するアプリケーションソフトもファイルを開くので，一度に開けるファイル数を多めに取っておかないといけません。したがって，CONFIG.SYS内のFILESの数を30以上に設定するのがよいでしょう（このことはマニュアルにも書いてあります)。

XDOSINIT. Xはシステムを立ち上げたとき1回実行すればいいので，AUTOEXEC.BATを書き直してXDOSINIT. Xを自動的に実行するのがいいでしょう。

ここまできたら，試しにリセットスイッチを押してみます。なるほど，うまくいきました。実際にはXDOS.Xなどのファイルはどこに置いてもかまいません。ただPATHが通っていれば，ディレクトリの指定が不要になります。

つまり，XDOS.Xで実行するプログラムがあるところにPATHが通っていると，ディレクトリやドライブの指定は必要ありません。これは非常に便利です。先ほどのTURBO PASCALの例でいくと，PATHがTURBO PASCALのところを通っていると現在のディレクトリやドライブに関係なく，

XDOS TURBO

で実行できることになります。

ただ忘れてならないのは，XDOS.XはMS-DOSのソフトを実行しますので拡張子（エクステント）が*.EXEか*.COMの実行可能ファイルしか実行できないということです。また，CONCERTOでは同名で*.EXEのファイルと*.COMのファイルが同一ディレクトリにあるときは*.COMのほうが優先して実行されます。*.EXEのほうを実行したいときは，ファイル名の指定のときにそのこともいえば*.EXEのほうが実行されることになります。

しかし実行時に，いちいちXDOSと入力するのも面倒な気がします。そんなときはバッチファイルを使う方法と，あの有名なMS-DOSのコマンドシェルのCOMMAND.COMを立ち上げる方法とがあります。

まず第1にバッチファイルを使う方法で

すが，これはバッチファイル内でXDOSなになにという命令を書いておくというものです。これはタイプする文字を少なくすることができますが，バッチファイルをそれぞれ作っておかないといけないので本質的な解決とはいえません。そこでCOMMAND.COMを使うのが適当といえます。COMMAND.COMを立ち上げるには，

XDOS COMMAND

と入力します。ただし，COMMAND.COMがカレントディレクトリになくPATHが通っていないときには，そのディレクトリを指定する必要があります。これはHumanと同様です。しかし，COMMAND.COMを立ち上げる前にやらなければいけないことがあります。それはXDOS.CNFファイル内のCOMSPEC環境変数に，COMMAND.COMのあるディレクトリを指定することです。たとえばCOMMAND.COMがルートの下のBINディレクトリにあったとすると，以下のように指定します。

SET COMSPEC=¥BIN¥COMMAND.COM

このように設定しておけば，以後 COMMAND.COMが立ち上げられるようになります。しかしCOMMAND.COMはVer2.1程度のものしか使えません。間違ってもVer.3.1のものは使えませんので注意してください。

さてこのXDOS.CNFファイルはこのほかにいろいろなことを書いておくことができます。そしてこのファイルの内容は COMMAND.COMが立ち上がったときに自動設定されます。たとえばプロンプトの指定です。現在使用中のシェルがCOMMAND.Xか，COMMAND.COMかが，ひと目でわかるようになります。ほかには環境変数の設定をけっこう自由にできます。これは，

SET 環境変数名＝環境変数の内容

ということになります。

また逆にCOMMAND.Xからの環境変数の継承をしないようにすることもできます。このためには，

DELENV

と書いておきます。すると，このDELENV以前に設定された環境変数は立ち上がったCOMMAND.COMにおいては破棄されます。このような環境変数に対する操作はきわめて重要であると思います。たとえば，XCとMS Cの両方を使いたいと思ったとします。するとこれらのCは環境変数によってコンパイルに必要なファイルの位置を指定します。ところが，両方の環境変数の名前がたまたま同じなのです。こうなるとどちらかのコンパイラを使うたびに環境変数の設定をやり直す必要があります。しかしHuman68kが立ち上がったときにXCの環境変数を設定し，XDOS.CNFファイルの中でMS Cの環境変数を設定するようにすれば，COMMAND.XからはXCが，COMMAND.COMからはMS Cが使えるようになります。このようにXDOS.CNFファイルをうまく使うことで，MS-DOSのアプリケーションソフトと，従来からあるHuman68kのソフトの両方を無理なく使えるようになります。

ただ，COMMAND.COMを立ち上げると従来のHuman68kの拡張子＊.Xのファイルは実行できなくなってしまいます。どうしても＊.Xのファイルを実行したいときは，

EXIT

と入力して，COMMAND.COMを抜けてCOMMAND.Xに戻ってから，＊.Xのファイルを実行することになります。ただし，このときCOMMAND.COMで設定した環境変数はCOMMAND.Xには継承されません。これは当たり前のことですが，非常に重要なことです。MS-DOSのエディタがあまり期待できない場合， Human68kのエディタ(EDなど）を使わざるをえません。これはとても面倒くさいことになります。というのは，COMMAND.COMを使って MS-DOSのコンパイラを実行したとします。しかし，ここでバグがあったならばエディタを起動してプログラムの修正をするのですが，まずCOMMAND.COMを抜けてエディタを起動するということになります。うーんこりゃ面倒だ。ということで頻繁に行う作業は，できるだけCOMMAND.X側のバッチファイルを利用したほうがよさそうです。

さらに，こんなことができたりするのだ？

V30というCPUは8080のエミュレータモードを持っています。Z80ならともかく，いまさら8080なんてという方もいるかもしれませんが，世界の標準であるCP/Mでは8080で十分（日本の標準であるS-OSは無理ですが)。ということで，V30が動いてるとなれば，8080エミュレータモードをなんとか使えないかと思うのが人情だろうと思います（うーん，エミュレータのエミュレータになっちゃうな)。そこでCP/Mのソフトが動かないかと試してみたのですが，うまくいきそうです。

今回使用したものはACEという技術評論社から出ていたエミュレータで，CP/M-80とCP/M-86兼用のものですが，うまく動作しています。この手のソフトはパブリックドメインのものが多く，ちゃんと発売されているものについてはあまりよく知りません。ただしパブリックドメインのものでは，誰も使いそうにないファンクションコールをエミュレートしないものもありますので，もしかすると動かないソフトもあるかもしれませんので注意してください。

これはハードウェアエミュレータなので速く，クロックが桁違いな分だけ本物のCP/Mより速くなります。実際 CONCERTOで動かすと8MHzの8080があるようなもので，きわめて強力といえます。MZ/X1シリーズからX68000に乗り換えた人も多いでしょうから，手持ちのCP/Mソフトを倍のクロックで走らせてみるというのも魅力的でしょう。問題はCP/Mのソフトを，どうやって2HDのMS-DOSフォーマットにコンバートするかです。なんとかシリアルで送るか，CP/M-86のお世話になるか。

さらに，2つのマイクロプロセッサがあれば並列に動作させて高速化を図りたいというのも人情のような気がします。実際CONCERTOでは並列動作ができるような構造になっているようです。腕に自信のある人は，頑張ってみるといいでしょう。ヒントはドライバです。CONCERTOのV30をHuman68k側から使うようなドライバを設計するのがもっとも簡単で使いやすいのではないかと思います。ただハードの知識がけっこう必要なので，それなりの人はやらないほうが身のためです。

最後にX68000, PC-9801, CONCERTO

表1　ファイル名の制限

作成	使用	使用についての注意
Human68k	MS-DOS	MS-DOSでは，小文字のファイル名やディレクトリ名は使えません。
Human68k	CONCERTO-X68K	小文字は大文字に変換されますが，すべてのファイルが使用できます。
MS-DOS	Human68k CONCERTO-X68K	ファイル名，ディレクトリ名に"－"がない限り，すべて使用できます。
CONCERTO-X68K	Human68k MS-DOS	すべて使用できます。

でファイル名の制約がかなり違います。これはPC-9801からソフトを持ってくるときなどに十分に注意してください。これを表1に示します。特に注意してほしいことはX68000, CONCERTOではファイル名やディレクトリ名などに“-”が入るとそのファイルなどに対する操作ができなくなってしまうということです。この場合はデバッガのお世話になるしかありません。

要するに, 結局どうなのだろうか

はっきりいって，このCONCERTOは生粋のエミュレータといえると思います。というのは従来のHuman68kのソフトを生かしつつ, MS-DOSのソフトを動くようにしたエミュレータだからです。このCONCERTOの場合, MS-DOSのソフトを実行中にX68000のフロントプロセッサASK68Kが動くなど, Human68kとMS-DOSがX68000上で協力して動いているような感触をユーザーに与えます。外観を別にすると，きわめて優れたアプリケーションといえます。

一般にエミュレータというのは遅いものですが，面白いことにコンパイルなどをするときは10MHzのPC-9801より速いときがあります(80286のVXよりも)。これはX68000のディスクアクセスが非常に速いためで，多くのファイルをアクセスしようとするときに顕著に表れるようです。

このCONCERTOはMS-DOSの単なるエミュレータとしてはもちろん，X68000のユーザーにいろいろなものを提供してくれそうなアプリケーションではないかと私は思います。 (相馬英智)

CONCERTO-X68K　99800円
アクセス　☎03(233)0200

図1　プログラム例

```
:sub sub_dm : ' 実宛名印刷サブルーチン

    IF        not !lp_sw and !lr_sw
        GOSUB     dm_prn
        GOTO      dm_end
      ENDIF

    IF        !lr_sw = 0
        MOVE      "" TO $w1
        MOVE      "" TO $w2
        MOVE      "" TO $w3
        MOVE      "" TO $w4
        MOVE      "" TO $w5
        MOVE      "" TO $w6
        MOVE      "" TO $w7
        MOVE      "" TO $w8
        MOVE      "" TO $w9
      ENDIF

    MOVE      ( !lr_sw + 1 ) * 35 TO !off_set

    MOVE      $w3 ++ "〒 " TO $w3
    MOVE      $w3 ++ 郵便番号 TO $w3
    MOVE      $w3 ++ string(!off_set-len($w3)," ") TO $w3

    IF        len(住所)<31
        MOVE      $w5 ++ "   " ++ 住所 TO $w5
      ELSE
        MOVE      instr(住所," ") TO !temp
        IF        !temp=0 or !temp>30 or !temp<20
            MOVE      $w5 ++ "   " ++ mid(住所,1,30) TO $w5
            MOVE      $w6 ++ "   " ++ mid(住所,31,len(住所)-30) TO $w6
          ELSE
            MOVE      $w5 ++ "   " ++ mid(住所,1,!temp) TO $w5
            MOVE      $w6 ++ "   " ++ mid(住所,!temp+1,len(住所)-!temp) TO $w
6
          ENDIF
      ENDIF
    MOVE      $w5 ++ string(!off_set-len($w5)," ") TO $w5
    MOVE      $w6 ++ string(!off_set-len($w6)," ") TO $w6

    MOVE      $w8 ++ "      " ++ 氏名 ++ "  様" TO $w8
    MOVE      $w8 ++ string(!off_set-len($w8)," ") TO $w8

    IF        !lr_sw
        GOSUB     dm_prn
        MOVE      0 TO !lr_sw
      ELSE
        MOVE      1 TO !lr_sw
      ENDIF

:dm_end

    RETURN
```

これがCARD PRO-68K

DATA PRO-68K/CARD PRO-68K

シャープのPRO-68Kシリーズとして2種類のデータベースソフトが登場します。ひとつはリレーショナルデータベースのDATA PRO-68K，もうひとつはカード型データベースのCARD PRO-68Kです。まず最初にことわっておきますと, DATA PRO-68Kは東海クリエイトのスウイングの移植版，CARD PRO-68Kはダットジャパンの隼というソフトのスーパーバージョンと考えられるものです。

KamikazeやBUSINESS PRO-68Kに続く久々のビジネスソフトですが，今回はサンプル版から見た製品の概要をお伝えしましょう。

リレーショナルとは

まず，リレーショナルとはどういうことでしょうか。関係データベースなどと訳すとどうも抽象的でなにか難しい高度な処理をやっているように思えますね。これはデータの扱い方からこう呼ばれているのですが，データとデータの相互関係などといっても全然要領を得ませんので，とりあえずリレーショナルデータベースではデータを表のかたちで扱うんだと考えておくとよいでしょう。ひとつのデータには決まった数の項目が用意されており，一定のフォーマットで蓄積されたデータなら，任意の項目をキーにして検索やソートができるというソフト，これがリレーショナルデータベースです。

一方，リレーショナルデータベースの扱う表をそのままビジュアルに表現したのがカード型データベースと考えられます。

これらのソフトを用いることで，従来汎用言語で記述されていた多くの業務用プログラムを簡単に作れるようになり，実に経済的です。

DATA PRO-68K

まずはDATA PRO-68Kです。従来，この手のソフトは高価（十数万円）なものだったのですが, X68000用には58,000円とかなり安く発売されます。

処理部はインタプリタとなっており，BASICのデータ処理強化版といった雰囲気もあります。図1にDATA PRO-68Kで記述された住所録のソースプログラムの一部を掲載します。CARD PRO-68Kなら10分でできそうなデータフォーマットですが，DATA PRO-68Kでは1000行以上のプログラムを書かねばなりません。その代わり，かなり自由なフォーマットでデータを蓄えることができます。その気になれば，市販ソフトクラスのものまで記述できるのです。

命令や関数はかなり豊富で，LOCATE, PRINTといったお馴染みのコマンドからLINEやPATTERN，果てはINKEY$（リアルタイムキー入力！）やPLAYまであり，これならビジネスソフトだけでなく，ブロック崩しくらいは作れるのではないかと思わせるほどです。

CARD PRO-68K

CARD PRO-68Kは従来のカード型データベースの弱点であった数式処理や多重条件検索などの機能的な不備を補っており，かなり本格的なデータベースといえます。

マウス対応，マルチウィンドウとなっており，メニュー選択などの操作性もなかなかよろしいようです。カード型データベースソフトというとPC-9801版でいくつか発売されていますが，やはりマウスがオプションのマシンではこういったソフトの真価は発揮されないのではないでしょうか。

メニューはX68000では珍しい完全なプルダウン方式。Kamikazeに慣れている人はもちろん，誰でも簡単に使いこなせそうです。ちょこちょことデータフォームを作成し，データを入力してみましたが，特にマニュアルを必要とすることなく作業を進めることができました。このノリのよさは重要でしょう。 (中野修一)

新製品速報

シティソフトよりX 68000用日本語フロントプロセッサFIXERとマクロアセンブラCMA68 Kが発売されます。FIXERは操作性と機動性に重点をおいたもので予価28,000円，発売日未定。CMA68 KはCに近いマクロ表記や68010，68020の命令までサポートしたもの。価格は19,800円，7月発売予定。

シティソフト　☎06(926)1060

MIDIサウンドプログラミング

●MIDI活用テクニック●
●再掲載X1/X1turbo用MIDIボードの製作●
●MIDI対応シーケンサ発表●

パソコンはMIDIと出会って“楽器”となります。といってもパソコン本体で音を鳴らすわけではありません。MIDIとは電子楽器を制御するための世界統一規格です。現在ではほとんどあらゆる電子楽器がMIDIを備えており，これらの演奏を自動的に制御する機器はシーケンサと呼ばれています。MIDIによってパソコンはシーケンサ，すなわち楽器の頭脳となるのです。
X1では1988年3月号のMIDIボードの製作により，やっとその下準備ができあがったところです。これからはそれをもとに新しい世界を開いていかなくてはなりません。MIDIを得たことにより，パソコンミュージックはプロミュージシャンと同じレベルにまで近づくことができるようになったのです。これからは“音楽”というレベルでパソコンミュージックを語るべきなのでしょう。
今回，X1用のMIDIボードの製作記事を再掲載するとともに，MIDIをフル活用するためのハードからソフトまでの集中講座，および現行FM音源用MMLに準拠したMIDI対応16チャンネルシーケンサを発表します。
コンピュータミュージックを見つめれば，これが本来の姿だったのではないかと気づくことでしょう。

短期集中講座　MIDI活用テクニック

MIDIの基礎とボードの製作

Misawa Kazuhiko
三沢 和彦

今回から3回にわたって，MIDIの徹底講義を行うことになりました。なんといってもX1に楽器をつなぐためにはMIDIボードを用意しなければなりません。3月号で発表したMIDIボードを再掲載するとともに，MIDIの基礎事項をしっかりとおさえておきましょう。

デジタル楽器の時代

XシリーズのFM音源はステレオ8重和音で音質もよく，祝氏が発表したMMLを利用すれば質の高いミュージックプログラムを作ることができます。ただ，本体とFM音源ボードだけで十分かというと，楽器としての使い勝手は，まだまだのようです。
一方，現在のミュージックシーンはシンセサイザの爆発的普及で，デジタル楽器なしには語ることができません。特に日本のデジタル楽器は信頼性が高く，その技術は世界的にも注目されています。この春に来日したクラシック界の帝王カラヤン(80歳)も，娘がデジタルシンセを買うというので都内の楽器店に立ち寄り，自ら深い関心を示していたということです。もはや，デジタルとかシンセとかいう言葉を聞いただけで顔をしかめる音楽ファンは少ないことでしょう。
ところで，皆さんはこれらのデジタル楽器とFM音源などを持つパソコンの中身がほとんど変わらないということを知っていますか？　同じであれば，X1も音源モジュールとして市販のデジタル楽器と接続できるはずですね。実際に本誌(1987年9月号)ではX1にヤマハのミュージックキーボードをつなぐという記事もありましたが，なぜ，一般の楽器のようにケーブルをつなぐだけというわけにいかないのでしょう。実は，市販のデジタル楽器が簡単に何台も接続して演奏できるのは，MIDIという統一規格で結ばれているからで，X1シリーズも楽器として扱うためにはそのMIDIが必要となるわけです。
この連載では，MIDIとは何かから始めて，FM音源を持ったX1シリーズを楽器として活用するノウハウを解説していくことにします。

表1　規格概要

データフォーマット	ボーレート：31.25Kbps 調歩同期式 データ長：8ビット　スタート・ストップビット：各1ビット
電気的特性	5mAカレントループ 受信側にフォトアイソレータをもつ
コネクタ	5pinのDINコネクタ
ケーブル	シールドされたツイスト線で最長15m。シールドはピン2に接続し，送信側のピン2のみグランドに落とす

MIDIとは何か

デジタル楽器に目覚ましい発展が見られるのは，かのYMO（イエロー・マジック・オーケストラ）の功績が大きいでしょう。YMOについては説明など不要でしょうが，彼らはあの時代にたった3人（プラス数人のサポートメンバー）で大量の音源を操り，見事なオーケストレーションをやってのけたのです。
普通オーケストラというからには数十人は必要でしょう。が，YMOはシーケンサというコンピュータに音程や音符長といった演奏情報を記憶させ（当時ローランドのMC-8というコンピュータは1台で8人分の演奏を記憶できた），そのコンピュータを連動してオーケストラを作ったのでした。
彼らはまさにパイオニアでしたから，独自にシステムを組んでいました。その後，多くの楽器メーカーがデジタル化に着手した結果，異なるメーカーの機種の間でも連動させる必要が出てきたのです。
具体的には，同じ演奏データが異機種間

で利用できること，すべての楽器のテンポが同期していることなどが必要です。そのために国際標準規格として定められたのがMIDI(Musical Instrument Digital Interface) なのです(表1)。

MIDI というのはひと言でいえば，音楽情報の通信システムです。実際，MIDI で行っていることはRS-232Cによるシリアル通信と同じことです（あとでたっぷりと説明しますが，RS-232C・マウスボードCZ-8BM2と今月再掲載されるMIDIボードはほとんど同じ部品で構成されています)。ただし MIDI の場合には，通信データ（メッセージという）まで規格化されているので，そのメッセージ内容もしっかり理解しなければなりません。

MIDI の世界は奥深くて，一度に全体を説明しようとするとパニックに陥ってしまいます。そこで，とりあえず私たちホビーユーザーがMIDI を使ってどういうことができるか考えてみましょう。

図1は筆者が使っているシステムです。X1のほかにシーケンサという音楽専用のコンピュータがあり，演奏情報はこれに記憶させています。キーボードとリズムマシンはここでは音源として使っています。私のキーボードは8音まで独立して出せるうえに(8重和音も8種の異なる音色も可)，シーケンサは16パートまで持てるので，これだけでも多重演奏ができるのです。もちろん，キーボードとリズムマシンとは MIDI によって同期していますから，同時演奏ができるわけです。

なお，マルチトラックレコーダ (MTR) は，ここでは単なるミキサーと考えておいてください。詳しくは連載の最後の回で説明できると思います。

私はこのシステムで，キーボード，ギター，ベース，ドラムス，ボーカルの5人編成のバンド演奏をひとりで実現しています(他人に聴かせられるような作品は皆無なのだが)。

さて，ここでのX1の役割ですが，いまのところシーケンサのデータをフロッピーディスクに落とすファイラーとしてしか使っていません。もちろん，単にプログラムを組んでいないだけで，やる気になればいろいろなことが実現可能です。実用的なソフトウェアは，この連載を読み終えた皆さんと私とでサポートしていきましょう。

ところで，私の場合はこれまでも音楽活動に力を入れてきたので，機材もいろいろと揃っていますが，これから MIDI の世界に入る皆さんが手軽にできることはなんでしょう。まず最低限必要なのは，MIDI ボードです。これは自作以外に手はありません（実は，CZ-8BM2をMIDI用に改造する計画もあるので，うまくいけば近いうちに発表しましょう)。あともうひとつ，MIDI楽器が必要です。

MIDI入門記事の多くは，YAMAHAのTX81Zやローランドの MT-32といった音源モジュールを勧めていますが，私はあまり賛成できません。音源ならFM音源ボードで十分だからです。むしろ，楽器としての鍵盤をつないで初めて MIDI の意味があると思うのです。ですから，これから MIDI 楽器を揃えようとするなら，自分はキーボードなんて弾けないよという人も，まずは鍵盤付きの楽器を購入することをお勧めします。

また，最近は MIDI 付きのギターやブラスも出揃っているようですが，それらの多くは音源に制約があるので使い勝手が悪いかもしれません。それから，もうひとつ注意すべきことがあります。それは，マルチモードの使えるキーボードでなければ意味がないということです。なるべく最近の機種で8音色同時に出せるものを選んでください。異なる音色での多重演奏もまたMIDIの醍醐味なのですから。

入門システムとしては図2のようになります。これだけでも，できることは無限に近いほどあります。たとえば，

1) 外部キーボードでX1のFM音源を鳴らす。
2) X1のメモリに演奏データを入れて自動演奏をさせる。
3) 外部キーボードの音色をX1上でエディットする。

この連載でも，上記の3点を具体的な目標におくことにします。

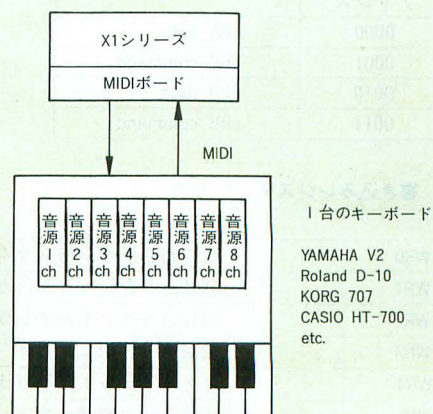
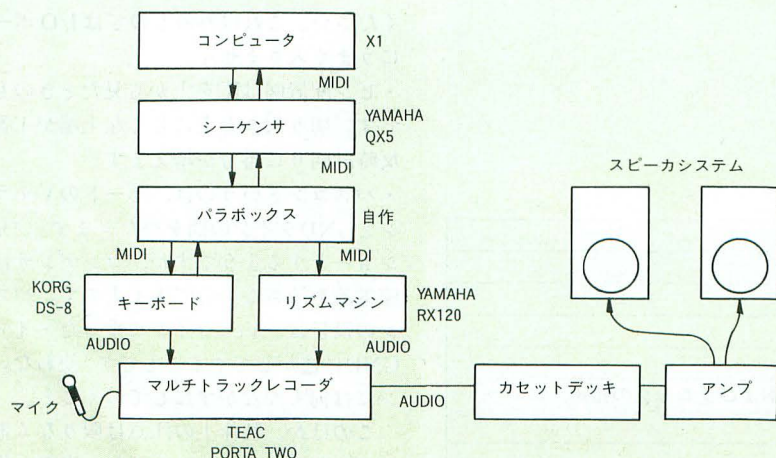
ではまず，最も基本となるハードウェア面の解説に入りましょう。

通信としてのMIDI

先ほど，MIDIはRS-232C と同じだと書きましたが，要するにどちらも Z80A SIO という石を使ったシリアル通信を行っているということなのです。だから,『試験に出るX1』の第5，6，7章をじっくりとながめると大部分のことがわかるはずです。とはいうものの，私はとても親切な人間なので，最初から説明してしまいましょう。

そもそもシリアル通信は，1本の通信線上を基準時間（ボーレート）ごとに1ビットずつHまたはLの信号を受け渡します。

図1 MIDIシステムの例

図2 お勧め入門システム

Z80のように8ビットデータの場合には，最下位ビットから最上位ビットが一定時間おきに送られますが，8ビットデータの始めと終わりが検出されなければなりません。そこで，スタートビット，ストップビットというのを付けることになります（これを調歩同期式という)。

もう一度，表1を見てください。MIDIではボーレートが31250ビット/秒（32μ秒間隔），スタート・ストップ各1ビットずつを足した計10ビットが1バイトデータの通信に必要で320μ秒かかります。

このシリアル通信を実現するハードウェアは，Z80A SIOを使えばたいへん簡単にすみます。このZ80A SIOというのは，もともとZ80 CPUにつなぐことを目的に設計されているので（これをペリフェラルという)，CPUの端子（すなわちI/Oポートの端子）とSIOの端子をほぼダイレクトに結線するだけでよいのです。

図6に3月号で発表したMIDIボードの回路図を再掲載します。初心者のために回路のそれぞれの部分の役割を説明しておきましょう。

まず，アドレスデコーダ部。LS136という石はXORというゲート（回路素子）が8個入ったものです。このXORを使ってAB2～AB9の値とDIP-SWの値がすべて等しいときだけLS138の入力AとBが“H”になるようにしています。

LS138は少しもったいない使い方をしているのですが，入力A，B，G1が“H”，入力C，G2A，G2Bが“L”のときだけ出力Y3が“L”になるようになっています。これは，SIOのCEに入っており，“L”のときだけSIOが動作します。これによって，I/Oアドレスは図4のように設定できます。アドレスの第1ビット，第0ビットはSIOのプログラミングに関係するものです。

クロック分周部は，31250ビット/秒のボーレートを作るところです。LS93で，Z80Aの4MHzのクロックを1/8にしてSIOのTxCA，RxCAに入力し，SIO内部でさらに1/16にしています。本当はこの分周にはZ80A CTCを使った回路のほうがよいのですが，それだと配線が多くなり，プログラミングも面倒になってしまいます。そこでLS93を使うことになったのです。

次は入出力部についてなのですが，特にMIDI INにフォトアイソレータ（PC900相当品：フォトカプラともいう）を使うことがMIDI規格に定められています。このフォトアイソレータは，電気信号の“H”と“L”をいったん光のON/OFFに換え，再び電気信号に戻すという一見面倒なことをしているのですが，こうすることによってMIDI INに接続した外部楽器と電気的に絶縁され，デジタルノイズが混入しないようになっているのです。

またMIDI THRUというのは，MIDI INの入力信号をそのまま外部に出しています。これを使えば次々と楽器を接続できますが，3台以上つなぐとMIDI信号の読み取りエラーが出る恐れもあります。楽器の数が増えてきたら，パラボックスという信号分配器を使う必要が出てきますが，これも自作が簡単なので，連載第3回目に製作記事を出そうと考えています。

図3　シリアル通信データ

図4　I/Oアドレス設定

AB15	AB14	AB13	AB12	AB11	AB10	AB9	AB8	AB7	AB6	AB5	AB4	AB3	AB2	AB1	AB0
0	0	0	0	0	0	*	*	*	*	*	*	*	*	B/A	C/D

DIP-SW設定値

通常はDIP-SWをすべて0にしておくので，

アドレス	
0000	chA data
0001	chB command
0010	chB data
0011	chB command

表2　書き込みレジスタ

	役目
WR0	次に書き込むレジスタを指定
WR1	割り込みを行うかどうかを指定
WR2	割り込みベクトルの指定
WR3	受信データ形式の指定
WR4	ストップビット，パリティビットおよびクロックの指定
WR5	送信データ形式の指定
WR6 WR7	調歩同期式では使わない

製　作

デジタル回路は，ICのピンをつなぐのが基本作業です。部品の配置が多少まずくてもとにかく番号を間違えずにつなぎさえすれば動くはずです。間違いをなくすために，1本つなぐたびに回路図の線をマーカーで消していくとよいでしょう。なお，表3も3月号からの再掲載です。

製作上の細かい注意を列挙しておくと，

・Z80A SIOは，バージョン0を使用すること。ほかにSIO/1，SIO/2がありますが，これらはピン配置が異なります。必ず買うときに確認してください。また，SIOには必ずソケットを使ってください。

・ボードは必ずX1用のMCC-153を使ってください。これ以外のものではI/Oポートにうまく入りません。

・ピン配置図はICを上から見たときのものです。切り欠きを上にして左上端が1番で，反時計回りに番号が増えます。

・パスコンというのは，ボードのVccラインとGNDラインの間をつなぎます。一見，ショートするようですが，コンデンサは直流電流を流さないので大丈夫です。パスコンの役目は，Vccラインに乗ったノイズをGNDに逃がしてやることです。使わないピンには何もつながずにしておくこと。

このほか，製作上の注意は限りなくあるので説明しきれません。もしも実際に作ってみて動かなければ，Oh!X「MIDIが動か

ないよ」係まで質問してください。ある程度たまったら誌上でお答えしましょう。

さて，配線が一応終了したら最初にすべきことは，まあ，お茶でも飲んで休憩することでしょう。最初は必ずどこか配線ミスをしているものですが，これを調べるのは頭を冷やしたあとのほうが効率がよいのです。

十分休憩して，配線チェックもOKとなれば，まずSIOを差さないでボードをX1本体に差し込み，電源を入れてみましょう。無事IPLが起動することを確認したら，必ず電源を落としてからボードを抜き，今度はSIOを差して改めてボードを装着し電源を入れてみます。私が作ったときは“Please turn on the Power SW slowly again！”というメッセージが表示されてしまいました。やはりミスがあったのです。誤配線がなければ必ず動作しますから，何度も根気よくチェックしてください。

SIOを差して，IPLが起動したら，このボードのMIDI INとMIDI OUTをケーブルでつないでループバックテストを行います。これは，その名のとおり自分で送信したデータを自分で受信してチェックするテストです。プログラムはリスト1のとおりです。

このリスト1のプログラムは，SIOを使ったプログラミングの最も基本的な部分ですので，少し詳しく説明してみましょう。

Z80A SIOのプログラミング

Z80A SIOはシリアル通信用の汎用LSIなので,MIDI用に利用するにはそれなりの初期設定が必要です。その内容は，SIO内部のレジスタに書き込んでやらなければなりません。書き込みレジスタはWR0～WR7の8個あります(表2)。今回は，WR0，1，3，4，5のみ理解すればOKです。

実際にレジスタに書き込むには，SIOのコントロールコマンドのI/Oアドレスにデータを出力します。

リスト1は，まず最初にこれらの初期設定を行っています。初期設定の内容はSIO INITルーチンの中で注釈をつけてあります。そしてこのプログラムは，1文字出力したいキャラクタをASCIIコードで送信し，自分自身で受信して合っているかどうかをチェックしているわけです。

ケーブルをつなぐときと，つながないときとで比較してみてください。また，ケーブルをつないでも合わない場合は，フォトアイソレータまわりの誤配線かケーブルのつなぎ違いあたりがいちばん怪しいところです。あるいはたまにあることですが，アドレスデコーダ部の誤配線，特にDIP-SWがすべてOFFになっていないかもしれません。もし，DIP-SWの設定を変えるならば，リスト1の60行でADRの値を変えてください。とにかく今月の目標は，このループバックテストにパスするところまでです。

ソフトウェアの実際

MIDIのソフトウェアを組むには，ハードウェアの知識だけでなくMIDIによって送受信されるメッセージのフォーマットについて理解しなければなりません。いうなればMIDI言語というものです。これについては来月詳しく解説しますので，今月は大まかなところだけ述べておきましょう。

MIDIボードでは，1バイト単位でデータを送受信しますが，MIDIメッセージは2バイトまたは3バイトをひとまとまりとしています。

1st	2nd	3rd
ステータス	データ	データ

1番目のバイトをステータスバイトといい，オペレーションコマンドを指定します。たとえば，発音（ノート・オン）なら90H，消音（ノート・オフ）なら80Hとなっています。

2番目以降はコマンドの中身を指示するデータバイトです。ノート・オン，ノート・オフなら2番目が音程，3番目が音の強さを表しています。

ステータスバイトは，最上位ビットが1（80H以上），データバイトは0（7FH），と決まっているので，ここで区別します。

このノート・オンとノート・オフだけでも，いじくってみると結構いろいろと遊べることと思います。たとえば，FM音源用MMLの発音，消音部分(KEY ON，KEY OFFルーチン）をMIDIメッセージのノート・オン，ノート・オフに差し換えただけで，これまでX1のFM音源を鳴らしていたMMLデータを使って，外部のMIDI楽器を

キーボードの選び方　ひと言アドバイス

街の楽器店には，機能も価格も実に多くの種類のキーボードが並んでいます。とはいえ，どのキーボードでも，使う目的にそった特徴がはっきりしているものです。ここでは，コンピュータミュージックの入門に適した機種の選び方を紹介します。まず最低限必要な条件は，

1） MIDI IN/OUTがあること。
2） 8音色同時発音が可能なこと。

最低限といっても，この2つを満たすキーボードはかなり限定されてきます。これをさらに，大きく分けると2つのタイプがあります。

A） 音色重視のシンセサイザタイプ
B） 手軽なポップキーボードタイプ

Aは，音源のパラメータが豊富で，凝った音づくりができます。弦楽器，管楽器の音は厚みや深みのある多彩な音が楽しめますが，唯一打楽器が苦手です。FM音源もこのタイプに属します。音色にうるさい人にはお勧めです。

Bは，音づくりはあまり凝ったことができませんが，そのかわりPCM音源という打楽器音をあわせて載せています。また，このタイプには自動伴奏機能がついているものも多いので，これ1台でなんでもやってしまおうという人にはいいでしょう。

どちらのタイプも価格は10万円前後が手頃なところです。自分の好みのタイプを見極めて選びましょう。では，A，B両タイプの代表機種を紹介しておきます。

A）	YAMAHA	V2	118,000円
	KORG	707	99,800円
	Roland	D-10	128,000円
B）	YAMAHA	DSR-1000	118,000円
	CASIO	HT-3000	89,000円

あとは実際に楽器店に行ってじかに触ってみるのがいちばん。店員さんをつかまえて質問すれば必ず答えてくれるはずです。

*　　*　　*

この原稿の締め切り直前，とびっきりの新製品情報が入ってきました。YAMAHAからEOS（Entertainment Operating System）というシンセサイザが新しく発売となったのです。

このEOSは，前述のV2と音源関係は変わらないのですが，本体中にさらにデジタルシーケンスレコーダ（8トラック，約10,000音記憶）を搭載したモデルYS200が129,000円となっています。デジタルレコーダはX1のMIDIシステムで実現可能ですから，自分でプログラムしてみようという人は下位機種のYS100（YS200からレコーダだけ除いたもの）が110,000円と，これも機能から見て格安といえます。特にAグループのなかに加えたとすると，そのなかでは最もお勧めといってよいでしょう。

演奏することができるようにもなります。実例を出そうと思ったのですが，今回は特集ということでMIDI用の本格的なシーケンサがこのあとの金子氏の記事で発表されることになっているので，これはそちらのほうに譲ることにしましょう。

来月に向けて

今月はとにかく，ひとりでも多くの人にMIDIボードを自作していただくことが狙いです。応用が非常に広い割には，比較的簡単にしかも安く自作できるボードなので，この機会にぜひともチャレンジしてみてください。

来月は，もう少し実用的なソフトウェアを作るためのノウハウを紹介するつもりです。具体的には，外部キーボードによるMML用フロントプロセッサを考えています。また，サポートしていきたいソフトウェアの希望があればどんどんお寄せください。そして，私たちの手でMIDIの世界をS-OSに匹敵する一大勢力にしようではないですか。

参考文献

祝一平：試験に出るX1，日本ソフトバンク
額田忠之：Z80ファミリハンドブック，CQ出版社
楽器用通信手順MIDIの使い方，トランジスタ技術，1987年7，8，9月号，CQ出版社
COMUSIC Vol. 1，2，3，立東社
島田奈美写真集＋α，学習研究社

表3　部品表

部品	数量	価格
Z80A-SIO/0	×1	1,000円
40P DIPソケット	×1	150～200円
フォトカプラ		
PC900	×1	150～200円
TTL		
74LS04	×1	
LS07	×1	
LS93	×1	
LS136	×2	
LS138	×1	
ダイオード		
ISI588	×1	
抵抗 1/8 or 1/4W		20～70円
220Ω	×5	
270Ω	×1	
1KΩ	×2	
3.3KΩ	×8	
10KΩ	×6	
コンデンサ		
47μF(電解10V以上)	×1	
0.1μF(パスコン)	×4	
ディップスイッチ		
6P	×1	320円
コネクタ		
FAP-26-07.02B(フラットケーブル用)	×1 or 2	500円
FAS-26-17(圧着)	×1 or 2	500円
26芯フラットケーブル(30～50cm)	×1	250円
DIN5P	×3 or 6	100円
基板		
サンハヤト MCC-153	×1	3,000～4,000円
	部品合計	約 8,500円

リスト1　ループバックテスト

```
10 'SAVE "SIOTEST1.Bas
20 '
30 '    Z80A SIO LOOP BACK TEST
40 '             63.6.12.    K.MISAWA
50 '
60 ADR=0
70 SIOD=ADR : SIOC=ADR+1
80 GOSUB "SIOINIT"
90 '
100 CLS
110 S$=INKEY$(1) : S=ASC(S$)
120 IF S<>&H1B THEN GOSUB "CHECK" : GOTO 110
130 GOSUB "SIORESET"
140 END
150 '
160 LABEL "CHECK"
170 PRINT#0 S$+" ";
180 '
190 OUT SIOD,S
200 T=INP(SIOD)
210 '
220 T$=CHR$(T) : PRINT#0 T$;
230 IF T=S THEN PRINT"  --- OK!" ELSE PRINT"  *** MISS!"
240 RETURN
250 '
260 LABEL "SIOINIT"
270 FOR I=1 TO 9
280 READ D : OUT SIOC,D
290 NEXT
300 DATA &H18 :'clear
310 DATA 1    :'WR1
320 DATA 0    :'interrupt disable
330 DATA 3    :'WR3
340 DATA &HC1 :'receive 8bit data
350 DATA 4    :'WR4
360 DATA &H44 :'stop bit=1bit,clock=1/16
370 DATA 5    :'WR5
380 DATA &H68 :'send 8bit data
390 RETURN
400 '
410 LABEL "SIORESET"
420 OUT SIOC,&H18
430 RETURN
```

図5　部品配置図

写真1

写真2

図6　回路図

X1 MIDI IF ver.1.0

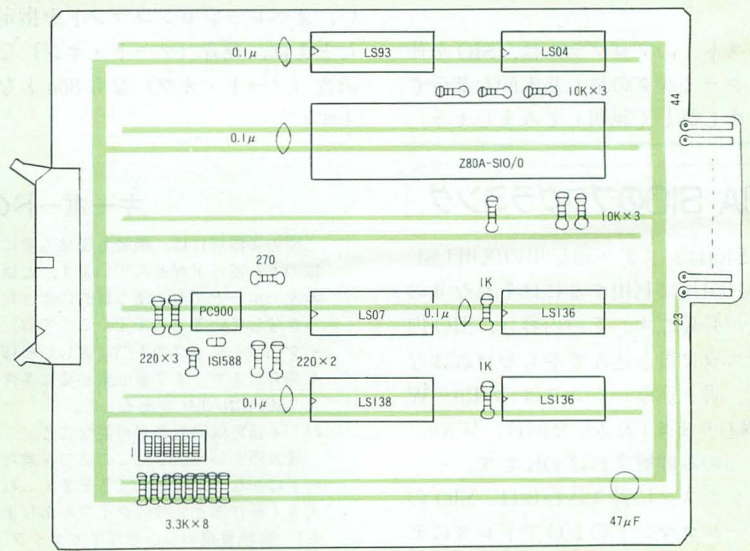
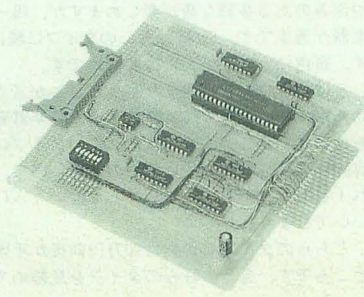
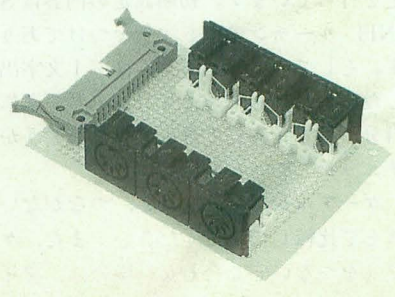
○各ICの電源及びGNDは省略
○SIO及びTTL 2個あたり1個にパスコン(0.1μF)を入れる

アドレスデコード部

入出力部

クロック分周部

26Pコネクタ
(偶数ピンはGND)

図7　ピン配置図

特集2 MIDIサウンドプログラミング

MIDI対応シーケンサ

Kaneko Shunichi
金子 俊一

MIDIを製作したら,次はそれを生かすソフトウェアが必要でしょう。というわけで,いよいよX1でMIDI楽器の演奏をコントロールする強力なシーケンサを発表できることになりました。もちろん,Oh!Xで標準となった祝版MMLに準拠したものとなっています。

夏ですね。どこかに遊びに行く計画はありますか? なになに,図書館に行くんですか。ひょっとしてあなた受験生? おお,同志よ,勝負はこれからですよ。がんばりましょうね。いやいや,V2するといぢめられますからね。

というわけで皆さん、今年の夏は音楽ですよ。それも気分を変えて自分で弾いてみるなんてのはどうでしょう。指が動かなくったって大丈夫。ちゃんとパソコンが面倒見てくれますよ。えっ,FM音源なら使ってるって? いえいえ,今回はもっといろいろな楽器を使ってみようというわけなんですよ。

そりゃ,FM音源の8重和音は強力ですし,PSGを足せば11重和音。それも,ゲームミュージックなら,アーケード版そっくりな音を作ることもできるでしょう。しかし,世の中には違うジャンルの曲のほうが圧倒的に多く存在するのです。そして,ピアノの音色が演奏者によって表情が違うのはいうまでもなく,ヴァイオリンの優しい息づかいや,そのぬくもりなど,それら数々の名音をOPMで忠実に再現するのは大変です。

でも,あきらめてはいけません。今のあなたにはMIDIという強い味方がいるのですから。さらに,自分で言うのもなんですが,MIDI対応のMMLとしては史上最強といわれる今回のシーケンサがあれば,鬼に金棒,ヤマトに波動砲,犬も歩けば猫も歩く,というくらいの自信作なのです。

このMIDIシーケンサでは,最大16パートの演奏データをMMLで記述することができます。すべてのパートをMIDI楽器用に割り当てることができるだけでなく,OPMやPSGなども併用することができるのです。

MMLの書式は,基本的には祝版に準拠したものとなっていますが,さまざまな微妙な表現を可能にするコマンドも追加しています。

というわけで,さっそくリスト3のダンプリストを入力してほしいのですが,その前にひとつだけ。このプログラムはBASICを改造するかたちをとっていますが,祝版MMLと同等のフリーエリアを確保するためにラインルーチンをカットしています。安全のために,2A60Hを1Bに書き換えておいてください。

それでは,各コマンドの説明をしながら概要を見ていきましょう。

M　チャンネル指定

MIDIのチャンネルを指定するコマンドです。MIDIには1から16までのチャンネルがあり,それぞれ独立したパートの演奏情報を送ることができます。

このMIDIシーケンサではMMLの各パートごとに1から16までの任意のチャンネル番号を指定できるようになっています。もちろんバラバラのチャンネルを指定してもよいのですが,同じチャンネルをいくつ指定してもかまいません。同じ音色でポリフォニックな使い方をしたい場合や,数台の楽器を使うときに威力を発揮します。これは,使用する楽器によって使い方が異なりますので,使用する楽器のマニュアルをよく読んでください。

このMコマンドは,X1シリーズのFM音源やPSGに対しては無効となります。なお,TEMPO0を実行しても初期化は行いませんので注意してください。

I　音色指定

そして,音色を指定できなければ話になりません。このコマンドは祝版MMLでもお馴染みですが,機能の拡張がなされています。

MIDI楽器のプログラムチェンジ(音色番号)は0～127となっていますが,このMIDIシーケンサではI0をPSG,I1～40をFM音源(OPM)に割り当てています。これは,祝版MMLと音色データを共有することを第一に考えているからで,MIDI楽器の音色を選択する場合には100番ずらしたI100～227を指定するようになっています。

さて,このMIDIシーケンサでは,最大16音(パート)まであるわけですが,OPMを使う場合には1音目から8音目までのパートで音色(I1～40)を指定してください。また,PSGを選択する場合には9音目から11音目までのパートで指定(I0)します。もちろん,MIDI楽器の場合には任意のパートを使って音色指定(I100～227)ができます。なお,起動時には1音目から16音目まですべてMIDIが選択されています。TEMPO 0を実行してもPSG, OPM, MIDIの選択は保存されます。

ところで,皆さんはFM音源やPSGのほかにどんな音源を知っていますか? LA音源,CD音源,VM音源,AI音源,iPD音源など,いっぱいあるんですよね。ことピアノの音色になるとFM音源がかわいそうなくらいよいものもあるんです。どうしてもショパンの「別れの曲」を奏でたかった私はお財布の中身と相談してKAWAIのK1m (VM音源) を買いました。全般的に比べるとやはりダントツはAI音源でしょう。そういえば,西川善司さんはKORGのM1(AI音源)を買ったそうです。今度遊びに行きますね。

P　パンポット指定

これは,パンポットを動かす(音を出す位置を変える)コマンドですが,OPMに対しては従来どおり,P1で左,P2で右,P3で両方となります。MIDIではP8が中央になり,左右それぞれ7段階に分かれP1が最左翼,P15が最右翼となります。

しかし,何段階サポートされているかは楽器によって異なりますし,実はMIDI規格にはパンポットなんてありません。このプログラムではコントロールチェンジのあとに10を送ってパンポットとしています。この値が異なる可能性もありますので,自分の楽器のマニュアルを調べてから使用してください。私のK1mではパンポットは3段階しかありませんので,デフォルトとしてRoland MT-32など用のデータをセットしておきました。

なお,OPMはIコマンドによってリセットされるので注意が必要です(MIDIでも楽器によってはリセットされるかもしれません)。また,PSGにはパンポット機能はありません。念のため。

＿　相対ボリューム

これは新しく追加されたコマンドで,相対ボリュームと呼びます。これにより,トレモロ,アクセント,フェードイン,フェードアウトなどの技が簡単に使えるようになりかなり有効だと思います。16音別々

にデフォルト値をもつことができるのも便利ですし，それゆえに値を省略することも可能です。いままでVコマンドを多用しなければできなかった微妙な表現もすっきりとしたプログラムでできるでしょう。

使用例をあげると，

PLAY "V120¯3__"

で，この場合にはボリュームは123まで上がったあと117まで下がります。

また，"V124¯4"や"V5_6"などのように，最大値や最小値を超える場合は無効となり，それぞれV124，V5のままになります。

なお，MML起動時のデフォルト値は0です。また，TEMPO0を実行してもデフォルト値は初期化されません。

∧　コントロールチェンジ

このコマンドはMIDI規格のコントロールチェンジを送るためにあります。具体的にはアフタータッチやポルタメント，ソステヌート，オールノートオフ，オムニモード，etc.……と聞き覚えのあるものから，よくわからないものまで，その名のとおりにコントロールできます。

使い方はいたって簡単です。コントロールチェンジは2バイトを送るのですが，たとえば，PLAY"^123,0"とかPLAY"^7，120"などです。

¥　ピッチベンダー

これは，ピッチベンダーを動かすためにあります。MIDI規格では14ビット分確保されているのですが，YAMAHAのショルキーでは7ビット，DX-7クラスで9ビットしか使われていないのが現状です。このプログラムでは8ビットを操作できます。基準値は128です。

Z　直接データ送信

これさえあればなんでもできるはずのコマンドです。つまり，ただ演奏データをタレ流すコマンドなのですが，なんでも流せるというのはやはり無敵です(某F社のMIDI対応MMLはこのコマンドしかない。う～ん見事なまでの手抜きぢゃ)。でも，良心的な私はさらに機能を拡張して，Zコマンドを2通りの使い方ができるようにしました。

それは，完全タレ流しのZ(　)と，エクスクルーシブのZ[　]です。両者の違いはZ[　]は最初に$F0を送り，最後に$ F7

リスト1　サンプル曲「Please Please Me」　日本音楽著作権協会(出)許諾第8870651-801号

```
10 '  `Please Please Me`
20 '
30 '                    by S.Kaneko
40 '
50 '                    in 26th Jun. 88'
60 CLS 0 :TEMPO 0
70 '
80 '            Tone List
90 '
100 '    i130 = B-Dru,Symbal      i132 = S-Dru,C-hh
110 '    i131 = Rim,Tom           i125 = Bass
120 '    i122 = Hamonica          i124 = B-Guiter
130 '
140 "P-A" :"P-B1" :"P-B" :"P-B2" :"P-B" :"P-B3"
150 '
160 PLAY "Z(252)"        ' Stop  For Timing Clock User !
170 '
180  POKE &HAFFF,1       ' Play
190 END
200 '
210 LABEL "P-A"
220 a$=" m1i130v127L4"            :b$=":m2i132v122L8"
230 c$=":m3i131v127L8"            :d$=":m4i125v110L8"
240 e$=":m5i122V118L4 ^1,46"      :f$=":m5i122v103L4"
250 g$=":m5i122v85L4"             :h$=":m6i124v100L4"
260 i$=":m5i122v118L4"
270  "!"
280  POKE &HAFFF,0       ' Play Pause
290 '
300 PLAY "Z(250)"        ' Start For Timing Clock User !
310 '
320 a$="t138o2 r"+STRING$(8,"cr")        'B-Dru
330 b$=":o5 r4"+STRING$(30,"c")          'HiHat
340 c$=":o4 r4m2"+STRING$(3,"r4ccr4cr")+"r4ccr4c16c16c"   'S-Dru
350 d$=":o3 b4>e4ee eeee eeee ee ^1,30_18<b16&+_>c16&+_c+16&+_d1
6^1,0¯¯¯¯ eeee eeee eeee ee ^1,30<b16&+_>c16&+_c+16&+_d16^1,0¯¯¯
"
360 e$=":o5 r4"+STRING$(2,"r4ed+c+<b.>c+8<bg+>")
370 h$=":o4 b e>ed+c+ <b.>c+8<bg+ r>ed+c+<b.>c+8<bg+"
380 f$=":" :g$=":" :i$=":"
390  "!" '0-4
400  LABEL "P-B"
410 a$=" o2 crcr crcr"
420 b$=":o5"+STRING$(16,"c")
430 c$=":o4 r4ccr4cr r4ccr4cr"
440 d$=":o4"+STRING$(16,"e")
450 e$=":o5 ^1,20 red+c+<b>c+<bg+"
460 f$=":o5 reee _2eeee"
470 g$=":" :h$=":" :i$=":"
480  "!" '5,6
490 a$=" o2 crcr c"
500 b$=":o5 cccc cccc c"
510 c$=":o4 r4ccr4cr r2 c16c16c16c16 c16c16c"
520 d$=":o4 aaaa eeee eg+g+a abbb"
530 e$=":o5 c+4.d+16c+16<b2"
540 f$=":o5 _e2¯e2"
550  "!" '7,8
560 a$=" o2 crcr crcr"
570 b$=":o5"+STRING$(16,"c")
580 c$=":o4 r4ccr4cr r4ccr4cr"
590 d$=":o4"+STRING$(16,"e")
600 e$=":o5 ^1,20 red+c+<b>c+<bg+"
610 f$=":o5 ¯reee _eeee"
620  "!" '9,10
630 a$=" o2 crc"
640 b$=":o5 cccc cc"
650 c$=":o4 r4ccr4c16c16c16c16 c4"
660 d$=":o4 aaaa eeee er¯2ee>ee<bb_"
670 e$=":o5 c+4.d+16c+16<b2 r2r4.b8"
680 f$=":o5 _e2¯e2"
690 h$=":o4 _5r1 re8e8>e8e8<b8b8"
700  "!" '11,12
710 a$=" o2 cr8c8cr" :a$=a$+a$
720 b$=":o5 cr4c cccc cr4c cccc"
730 c$=":o4 rc16c16cr4.cc rc16c16cr4.cc"
740 d$=":o4 a4rer2 f+4rc+4.fg"
750 e$=":o5 c+r2r8c8 c+r2r8c8"
760 f$=":o5 re8e.r rf+8f+."
770 g$=":o4 ra8a.r r>c+8c+.r"
780 h$=":o5 c+2.r8c8c+2.r8c8"
790  "!" '13,14
800 d$=":O4 g+4rc+4.g+c+ a4re4.a4"
810 e$=":o5 c+r2r8d+8 er2<b"
820 f$=":o5 rg+8g+.r ra8a.a"
830 g$=":o5 rc+8c+.r rc+8c+.<b"
840 h$=":o5 c+2.r8c8 c+2.¯"
850  "!" '15,16
860 a$=" o2 crcr crc"
870 b$=":o5"+STRING$(16,"c")
880 c$=":o4 r4ccr4cr r4ccr4c"
890 d$=":o4 eeee eeee aaaa bbbb"
```

```
900 e$=":o5 b2g+f+8&+g+8 e.e8d+d+"
910 f$=":o5 g+2g+g+ er8e8d+d+"
920 g$=":o4 b2bb >c+r8c+8<bb"
930 h$=":"
940  "!" '17,18
950 RETURN
960  LABEL "P-B1"
970 b$=":o5"+STRING$(14,"c")
980 c$=":o4 r4ccr4cr r4ccr4c16c16c"
990 d$=":o4 eeeg+ bbg+e aaaa bb<bb"
1000 e$=":o4 b2"
1010 f$=":o4 b2"
1020 g$=":o4 g+2"
1030 h$=":o5 red+c+ <b.>c+8<bg+"
1040 i$=":o5 r^1,46ed+c+<b.>c+8<bg+"
1050  "!" '19,20
1060 RETURN
1070  LABEL "P-B2"
1080 a$=" o2 crcrc"
1090 b$=":o5"+STRING$(9,"c")
1100 c$=":o4 r4ccr4cr rc16c16c _7m3>e16e16ee16e16ee~" 'S-Dru > T
om
1110 d$=":o4 eeeg+ bbg+b e"
1120 e$=":o4 b2"
1130 f$=":o4 b2"
1140 g$=":o4 g+2"
1150 h$=":o5 red+c+ <b8"
1160 i$=":o5 r^1,46ed+c+<b"
1170  "!" '21,22
1180 a$=" o2 cccc cccc"
1190 b$=":o5"+STRING$(16,"c")
1200 c$=":o4 m2c4c4c4c4 c4c4c4c4" ' S-Dru
1210 d$=":o4 aaf+4e4c+4 b4a4g+4f+4"
1220 e$=":o5 L8 ^1,20c+<b+>c+<b+>c+<b+>c+c+ d+c+d+c+d+c+<bb"
1230 f$=":o5 _e1_f+1~"
1240 g$=":o4 _a1_b1~"
1250 h$=":" :i$=":"
1260  "!" '23,24
1270 a$=" o2 crcr crc"
1280 b$=":o5 r1 cccc cccc"
1290 c$=":o4 c16cc16cc _m3>e16ee16ee~ m2r4cr4ccr" ' S-Dru > Tom
1300 d$=":o4 e4eg+b4g+b e4eg+b4g+e"
1310 e$=":o5 c+4.d+16c+16<b2"
1320 f$=":o5 _e1~ re8.<b16>c+<b"
1330 g$=":o4 _b1~ rb8.g+16ag+"
1340  "!" '25,26
1350 a$=" o2 cccc cccc"
1360 b$=":o5"+STRING$(16,"c")
1370 c$=":o4 crcr cccr crcr cccr"
1380 d$=":o4 a4f+4e4c+4 b4bag+4f+4"
1390 e$=":o5 c+<b+>c+<b+>c+<b+>c+c+ d+c+d+c+d+c+4<b"
1400 f$=":o5 e1~f+2.r8<b8"
1410 g$=":o4 a1~b2.r8b8"
1420  "!" '27,28
1430 d$=":o4 e4eg+b4g+e a4aab4bb"
1440 e$=":o5 b2.f+g+ e4.ed+d+c+<b"
1450 f$=":o5 g+2._g+ e.e8f+8f+8f+8e8"
1460 g$=":o4 b2._b a.a8b8b8b8g+8"
1470  "!" '29,30
1480 d$=":o4 e4eg+b4g+e a4a4b4<b4"
1490 e$=":o4 b2L4"
1500 f$=":o5 e2"
1510 g$=":o4 g+2"
1520 h$=":o5 red+c+ <b.>c+8<bg+"
1530 i$=":o5 r^1,46ed+c+<b.>c+8<bg+"
1540  "!" '31,32
1550 RETURN
1560  LABEL"P-B3"
1570 d$=":o4 eeee eeee aaaa bbbb"
1580 e$=":o4 b2r>f+8g+8 e.e8d+d+~2"
1590 f$=":o5 b2g+f+8g+8 e.e8d+d+~1"
1600 g$=":o4 g+2rb >c+.c+8<bb~2"
1610 h$=":o5 red+c+ <b.>c+8<bg+"
1620 i$=":o5 red+c+ <b.>c+8<bg+"
1630  "!" '33,34
1640  "!" '35,36
1650 a$=a$+"c1"
1660 b$=":o6 _5m1c2c2 c2c2 c1" 'Synbal
1670 c$=":o4"+STRING$(2,"r4c16c16c16c16 r4c16c16c16c16")
1680 d$=":o4 ~2e4r4g4r4 c4r4<b4r4 >e1"
1690 e$=":o4 b2"
1700 f$=":o5 b0&b2."
1710 g$=":o4 g+2"
1720 h$=":" :i$=":"
1730  "!" '37,38
1740 RETURN
1750 '
1760 END
1770  LABEL "!"
1780 PLAY a$;:PLAY b$;:PLAY c$;:PLAY d$;:PLAY e$;:PLAY f$;:PLAY
g$;:PLAY h$;:
1790 PLAY i$;:PLAY j$;:PLAY k$;:PLAY l$;:PLAY m$;:PLAY n$;:PLAY
o$;:PLAY p$
1800 RETURN
```


を送るということです。たとえば，Z(240, a, b, c, 247)とZ[a, b, c]は同じ動作をします。なお，エクスクルーシブの$F0の次のデータは楽器メーカーごとに決まったIDナンバーを送ることになっているので注意してください。

S　CTC制御

超マニア向きのテンポコマンドです。CTCを直接触ることになるので，はっきりいって初心者の方(CTCについて理解していない人)は使わないでください(どんなことが起こっても私は責任を持てません)。

どうして，こんなコマンドを用意したかというと，微妙なテンポの変化を味わうためなんです。クラシックや，特にピアノ曲を奏でたいときに，本気でMML化するなら，なんとしても欲しくなってくる機能なのです。

では，そのようなマニアックな人のために，このSコマンドについて簡単に説明しておきましょう。ただし，ここではCTCにはチャンネルが4つあることは常識とします。このMIDIシーケンサでは，CTCの4つのチャンネルのうち，チャンネル0をタイマモードで，チャンネル3をカウンタモードで使っています。

Tコマンドでは，チャンネル3の値を変えて割り込みのタイミングを得ていますが，チャンネル0の値が結構おおまかなため，テンポがかなりずれることになります。この点，Sコマンドではチャンネル0もチャンネル3も直接指定できるので，限界までいじることができます。この場合Sチャンネル3，チャンネル0のようにデータをセットしてください。

チャンネル0は省略可能です。理論上，祝版MMLのTコマンドの16倍の細かさまで設定できます(単にプリスケーラの問題なのだが)。たとえば，T120＝S30, 43(L1＝@384)になります。

POKE &HAFFF,0

この命令で，曲データを演奏せずにバッファリングすることができます。今回のシーケンサでは，フリーエリアは16Kバイト強しかあいていません。これでは48Kバイトもの演奏用ワークエリアを有効に使うことは難しく，これまで発表されてきたミュージックプログラムのなかにもメモリで苦労しているものがかなりありました。16チャンネル同時発声が可能になった今回のシ

ーケンサでは，さらにメモリ不足が予測されるので，このコマンドが登場したわけです。

演奏開始は，POKE &HAFFF, 1 としてください。なお，TEMPO0により，1が書き込まれますので注意してください。

過去のMMLデータの演奏について

このMIDIシーケンサでは，全音符がL1=@384になっています。祝版MMLではL1=@1024なので，@コマンドで直接音長を指定しているプログラムでは注意が必要ですが，音長をセットするプログラム（リスト2)により解決されます。なお，リスト2ではMIDIのタイミングクロックのON/OFFを設定することもできますが，ONにできるのは音長が192と384の場合となります。

また，&と&+が，逆の機能になっているのも気をつけてください。ただし，組曲「イース」(3月号に掲載)ではもともと&と&+を逆にしているので，そのまま演奏できます。

あとはPOKE，PEEK 命令で何をしているかによって，アドレスをずらすなどの処理が必要です。さらに，PSGを使っている曲ではI0の指定も忘れずに。

N，Wコマンドもありませんので，それらを使っているMMLデータは演奏できません。自分で工夫してみてください。アフターバーナー(7月号)ではCTCを直接指定しているので，Sコマンドで対処する必要があります。ただ，かなりタスクが苦しいので，セットするデータを変えないと演奏できません。もしくは“T170”程度でほぼ同等のスピードになります。

Please Please Me

サンプル曲としてビートルズの「Please Please Me」を作ってみました(リスト1)。あくまでも私の愛用しているKAWAIのK1mを利用して作ったものなので，音色番号は自分の楽器の音色とよく相談して適当に変えるなり，自分でエディットするなりしてください。皆さんの音楽心に期待しています。

また，今回のシーケンサ（と呼ぶのも大げさだが）についてご意見のある方は編集室までお便りください。

リスト2　音長セットプログラム

```
10 '     MML Configuration Tool
20 '
30 '              by S.K
40 '
50  DEFINT A-Z :RESTORE 340 :POKE &HAFFF,0
60 FOR i=1TO4 :READ L(i),CTC(i) :P$(i)=STR$(L(i)):NEXT
70 FOR j=1TO2 :READ Q$(j) :NEXT
80 IF PEEK(&HAA5F) THEN j=1 ELSE j=2
90 L(0)=PEEK(&HAEFA)*256+PEEK(&HAEF9)
100 P$(0)=STR$(L(0)) :CTC(0)=PEEK(&HA970)
110 LTBL=&HAEF7 :i=0
120 '
130 PRINT" Length L1 =@";P$(i);" ";
140 Z$=INPUT$(1)
150  IF Z$=CHR$(13) THEN PRINT :GOTO 170 ELSE i=VAL(Z$)
160  IF i<1 OR i>4 THEN 140 ELSE PRINT CHR$(2,2,2,29); :GOTO130
170  IF i>2 THEN j=2
180 PRINT" Timing Clock ";Q$(j);" ";
190 Z$=INPUT$(1)
200  IF Z$=CHR$(13) THEN 220 ELSE j=VAL(Z$)
210  IF j<1 OR j>2 THEN 190 ELSE PRINT CHR$(2,2,2,29); :GOTO170
220 '
230 POKE &H4093,L(i)/96 :POKE &H4098,L(i)/96
240 POKE &HAA5F,0,0,0
250 IF j=2 THEN270
260 POKE &HAA5F,&HCD,&H85,&H40
270  POKE &HA970,CTC(i)
280  FOR k=1 TO 32
290   MEM$(LTBL+k*2,2)=MKI$(L(i)/k)
300  NEXT
310 MEM$(LTBL,2)=MKI$(L(i)*2)
320  POKE &HAFFF,1 :END
330 '
340 DATA 192,85,384,43,512,16,1024,16
350 DATA On,Off
```

リスト3　MIDIシーケンサ(ダンプリスト)

```
A8B0 21 3E A9 22 E3 2D 22 11 : 6D
A8B8 2E 22 13 2E 21 00 B0 11 : 73
A8C0 6E 3D 01 7A 03 ED B0 C9 : 8F
A8C8 00 00 00 00 00 00 00 00 : 00
A8D0 00 00 00 00 00 00 00 00 : 00
A8D8 00 00 00 00 00 00 00 00 : 00
A8E0 00 00 00 00 00 00 00 00 : 00
A8E8 00 00 00 00 00 00 00 00 : 00
A8F0 00 00 00 00 00 00 00 00 : 00
A8F8 00 00 00 00 00 00 00 00 : 00
A900 00 00 00 00 00 00 00 00 : 00
A908 00 00 00 00 00 00 00 00 : 00
A910 00 00 00 00 00 00 00 00 : 00
A918 00 00 00 00 00 00 00 00 : 00
A920 00 00 00 00 00 00 00 00 : 00
A928 00 00 00 00 00 00 00 00 : 00
---------------------------------
SUM: BD 9D BD CA 07 1A 82 EB 1ED9

A930 00 00 00 00 00 00 00 00 : 00
A938 00 00 00 00 00 00 CD D1 : 9E
A940 7F 3A DB A5 E5 EB FE 03 : 0A
A948 CA D4 A9 CD 36 54 CD AF : 1A
A950 5B 7C B5 20 4E CD 7F 3E : 84
A958 21 01 40 22 2A AA 22 2C : A6
A960 AA 2B 22 28 AA 44 4D AF : 09
A968 ED 79 3E 1E 32 D8 AE 3E : B8
A970 2B 32 D7 AE 32 FF AF 21 : E3
A978 49 AF 11 4D AF 01 40 00 : 46
A980 ED B0 EB 06 10 36 00 23 : F7
A988 10 FB F3 3E C3 32 3C 01 : 6E
A990 21 A5 A9 22 3D 01 CD 3C : D8
A998 01 CD 2E AA 21 4B AA 22 : DE
A9A0 5E 00 FB E1 C9 01 07 07 : 12
A9A8 3E 03 ED 79 16 07 3E 08 : 0A
---------------------------------
SUM: 8B 30 5E 5F 60 8E 1B 8C A00E
```

```
A9B0 CD 3F AE 15 20 FA CD 3F : F5
A9B8 AE 3E B0 F5 CD 9F 3E 3E : 79
A9C0 7B CD 9F 3E AF CD 9F 3E : 7E
A9C8 F1 3C FE C0 20 ED 01 00 : F9
A9D0 1C C3 3F 01 CD C3 7F B7 : E5
A9D8 28 03 CD F5 A9 E1 7E D6 : CB
A9E0 3B 23 C8 2B ED 4B 2C AA : 5F
A9E8 AF ED 79 03 ED 43 2C AA : 1E
A9F0 ED 43 2A AA C9 ED 4B 2C : 31
A9F8 AA 2A 2A AA B7 ED 42 20 : AE
AA00 05 2E 3A ED 69 03 6F 1A : 4F
AA08 FE 61 38 06 FE 7B 30 02 : 48
AA10 C6 E0 ED 79 03 13 78 B1 : 4B
AA18 28 08 2D 20 EA ED 43 2C : C3
AA20 AA C9 CD 3C 01 C3 27 20 : 87
AA28 00 40 00 40 00 40 01 04 : C5
---------------------------------
SUM: 47 49 F5 88 E1 E0 0F 05 C1E9

AA30 07 3E 07 ED 79 3A D7 AE : 71
AA38 ED 79 3E 58 ED 79 01 07 : 6A
AA40 07 3E C7 ED 79 3A D8 AE : 32
AA48 ED 79 C9 F3 ED 73 3C A9 : 67
AA50 31 3C A9 F5 C5 D5 E5 21 : AB
AA58 FF AF 7E B7 CA D3 AA 00 : 2A
AA60 00 00 21 8D AF 06 10 AF : 22
AA68 B6 23 20 51 10 FA ED 4B : 8C
AA70 28 AA 2A 2A AA 2B B7 ED : 9F
AA78 42 20 07 3E 01 32 FF AF : 88
AA80 18 51 ED 78 03 28 0C ED : F2
AA88 78 03 20 FB 0B ED 43 28 : F9
AA90 AA 18 DF ED 43 28 AA 21 : C4
AA98 8D AF E5 21 9D AF 16 00 : A4
AAA0 ED 78 28 05 03 FE 3A 20 : ED
AAA8 F7 B7 28 02 3E 01 E3 77 : 71
---------------------------------
SUM: E3 90 8F 9F F4 50 5A 90 5DFE
```

```
AAB0 23 E3 71 23 70 23 14 7A : BB
AAB8 FE 10 38 E4 E1 11 00 00 : 1C
AAC0 21 8D AF D5 E5 7E B7 C4 : 10
AAC8 DE AA E1 D1 23 1C 7B FE : F2
AAD0 10 20 F0 E1 D1 C1 F1 ED : 71
AAD8 7B 3C A9 FB ED 4D E5 D5 : 4F
AAE0 F2 F5 AA CD 4A AE 78 B1 : 7F
AAE8 D1 E5 CC 36 AD E1 7C B5 : 77
AAF0 E1 C0 36 01 C9 CB 23 21 : B0
AAF8 9D AF 19 4E 23 46 ED 78 : 81
AB00 28 04 FE 3A 20 08 D1 70 : CD
AB08 2B 71 E1 36 00 C9 D1 ED : 3A
AB10 78 28 F4 FE 3A 28 F0 D5 : B9
AB18 E5 CD 29 AB E1 28 E7 D1 : 47
AB20 38 ED 70 2B 71 E1 36 80 : C8
AB28 C9 ED 53 FD AF ED 78 03 : 1D
---------------------------------
SUM: 9D 13 56 1C 55 6B 47 83 A8A6

AB30 FE 41 38 2A FE 48 30 34 : 4B
AB38 CD 49 AB C5 D5 CD B1 AC : 85
AB40 D1 C1 CD 4E AD 3E 01 B7 : 50
AB48 C9 D6 41 87 6F ED 78 03 : 3E
AB50 FE 2D C8 2C 2C FE 2B C8 : 3C
AB58 FE 23 C8 2D 0B C9 FE 3E : 26
AB60 CA 6B AC FE 3C CA 6B AC : FC
AB68 7F B7 37 C9 FE 52 28 D2 : 80
AB70 FE 49 CA 6E 3D FE 5A CA : DE
AB78 47 3F FE 5E CA 73 3F FE : 5C
AB80 5C CA 9C 3F FE 4D CA 09 : 1F
AB88 3F FE 54 CA 7D AC FE 59 : DB
AB90 CA 3D AC FE 4B CA 1D AC : 8F
AB98 FE 7E CA 98 AE FE 5F CA : B3
ABA0 78 AE FE 53 CA BD 3F FE : 3B
ABA8 50 28 39 21 00 00 FE 4F : 1F
---------------------------------
SUM: 1A 74 C9 C3 A5 12 30 0B 41F7
```

```
ABB0 28 0F 2C FE 56 28 0A 2C : 15
ABB8 FE 51 28 05 2C FE 4C 20 : 12
ABC0 A7 E5 CD CC AE D1 19 E5 : A2
ABC8 F5 CD FA AD 30 04 F1 D1 : 5F
ABD0 18 96 67 F1 D1 EB 73 FE : 33
ABD8 4C 20 8D 7A FE 02 20 88 : 1B
ABE0 CB FE 18 84 CD 02 40 CA : 3E
ABE8 DC 3F 7B FE 08 D2 68 AB : 81
ABF0 C5 D5 CD FA AD 38 21 7D : E4
ABF8 FE 04 30 1C D1 D5 D5 F5 : BE
AC00 21 39 40 19 6E CD 5B 3E : 87
AC08 7E E6 3F 4F F1 0F 0F B1 : B2
AC10 D1 57 3E 20 83 CD 3F AE : C3
AC18 D1 C1 C3 68 AB 7B FE 08 : E9
AC20 D2 68 AB C5 D5 CD FA AD : F3
AC28 D1 38 0E 7D FE 40 30 09 : 0B
---------------------------------
SUM: 74 B5 D8 B1 E2 FA 62 CA 38E8

AC30 07 07 57 3E 30 83 CD 3F : 62
AC38 AE C1 C3 68 AB D5 CD FA : E1
AC40 AD 38 24 ED 78 FE 2C 20 : B8
AC48 1E 03 E5 CD FA AD 7D E1 : D8
AC50 38 15 D1 C5 57 7B FE 08 : BB
AC58 7D 38 07 CD 2D AE C1 C3 : E8
AC60 68 AB CD 3F AE 18 F7 D1 : AD
AC68 C3 68 AB D6 3D CD CC AE : 30
AC70 86 FA 68 AB FE 09 D2 68 : D4
AC78 AB 77 C3 68 AB CD FA AD : 6C
AC80 DA 68 AB 7D FE 1E DA 68 : C8
AC88 AB C5 4D 06 00 11 4E 0E : 30
AC90 21 00 00 3E 10 CB 23 CB : 28
AC98 12 ED 6A B7 ED 42 30 02 : 81
ACA0 09 1D 1C 3D 20 EF 7B 32 : 3B
ACA8 D8 AE CD 3E AA C1 C3 68 : 27
---------------------------------
SUM: 2A B9 E9 0D 2A D3 4A 76 C860

ACB0 AB CD 02 40 CA 2F AD 7B : DB
ACB8 FE 08 38 3A 7D D5 CD CC : 63
ACC0 AE 46 23 4E 87 5F 21 D9 : 45
ACC8 AE 19 56 23 5E CB 3B CB : 6F
ACD0 1A 10 FA E1 61 7D D6 08 : C1
ACD8 87 CD 2D AE 3C 53 CD 2D : B8
ACE0 AE 7D 54 CD 2D AE 3E 07 : 6C
ACE8 16 F8 CD 2D AE 3E 0D CD : CE
ACF0 36 AE CD 2D AE C9 7D B7 : 89
ACF8 20 01 3D C6 17 FE 1B 38 : 8C
AD00 0B D6 10 FE 0F 38 05 FE : 39
AD08 12 28 01 3C D5 CD EB 3D : 41
AD10 F5 CD CC AE 7E 3D 87 87 : 05
AD18 87 87 57 F1 82 57 E1 7D : 8D
AD20 C6 28 CD 3F AE 7D F6 78 : 93
AD28 57 3E 08 CD 3F AE C9 CD : ED
---------------------------------
SUM: 76 ED 0E 4C 3A 75 73 67 0F7C

AD30 EA 3E CD B0 3E C9 CD 02 : 7B
AD38 40 CA C9 3E 7B FE 08 38 : CA
AD40 06 16 00 CD 2D AE C9 53 : E0
AD48 3E 08 CD 3F AE C9 D5 CD : 6B
AD50 65 AD 2B 1B ED 53 63 AD : A8
AD58 D1 C5 ED 4B 63 AD CD 66 : 11
AD60 AE C1 C9 00 00 CD CC AE : 7F
AD68 23 23 56 23 5E D5 ED 78 : 57
AD70 FE 40 20 21 03 11 00 00 : 93
AD78 CD 1A AE 30 05 11 01 00 : DC
AD80 18 41 5F CD 1A AE 38 3B : C0
AD88 62 6B 29 29 19 29 5F 16 : D6
AD90 00 19 EB 18 EE CD FA AD : 7E
AD98 30 09 6B CB 7D 28 04 CB : E3
ADA0 BD 3E FE F5 7D FE 21 38 : C2
ADA8 02 2E 20 F1 26 00 CB 25 : 57
---------------------------------
SUM: A9 10 64 93 8B CC DE B9 3889

ADB0 11 F7 AE 19 5E 23 56 FE : A4
ADB8 FE 20 08 62 6B CB 3A CB : C3
ADC0 1B 19 EB 62 6B F1 E5 67 : 29
ADC8 CD 0B 40 ED 78 FE 26 7C : 1D
ADD0 E1 20 13 03 ED 78 03 FE : 7D
ADD8 2B 28 1D 0B 21 FF FF 3E : D8
ADE0 01 CD 0C 40 18 12 3D 28 : A9
ADE8 03 19 18 FA CB 3C CB 1D : 1D
ADF0 CB 3C CB 1D CB 3C CB 1D : DE
ADF8 EB C9 ED 78 28 1A FE 3A : 93
AE00 28 16 CD 1A AE D8 6F CD : E7
AE08 1A AE 38 0A 67 7D 87 87 : FC
AE10 85 87 84 6F 18 F1 B7 C9 : 88
AE18 37 C9 ED 78 FE 3A 38 02 : D7
AE20 37 C9 D6 30 03 D0 FE FE : D5
AE28 28 01 0B 37 C9 01 00 1C : 51
---------------------------------
SUM: 1A 4C 44 19 87 49 51 BD 67B9

AE30 ED 79 05 ED 51 C9 01 00 : 73
AE38 1C ED 79 05 ED 50 C9 C5 : 52
AE40 01 00 07 ED 79 0C ED 51 : B8
AE48 C1 C9 D5 CB 23 CB 23 21 : 5C
AE50 41 40 19 4E 23 46 23 5E : D2
AE58 23 56 EB 0B 2B D1 C5 E5 : 15
AE60 CD 66 AE E1 C1 C9 CB 23 : 3A
AE68 CB 23 E5 21 41 40 19 71 : FF
AE70 23 70 23 C1 71 23 70 C9 : 44
AE78 CD FA AD 38 05 7D CD C7 : C2
AE80 AE 77 CD CC AE 23 7E E5 : F2
AE88 CD C7 AE 96 38 06 E1 77 : 6E
AE90 3C C3 68 AB E1 C3 68 AB : C9
AE98 CD FA AD 38 05 7D CD C7 : C2
AEA0 AE 77 CD 02 40 3E 7F 28 : 19
AEA8 09 7B FE 08 3E 7F 38 02 : 81
---------------------------------
SUM: F2 A5 1C 4D EA D6 2E 96 E7C4

AEB0 3E 0F CD CC AE 23 E5 F5 : 91
AEB8 7E CD C7 AE 86 5F F1 BB : 51
AEC0 38 D2 E1 73 C3 68 AB 21 : 55
AEC8 39 AF 19 C9 D5 21 4D AF : BC
AED0 CB 23 CB 23 19 D1 C9 00 : 8F
AED8 00 D0 12 C2 11 C4 10 D2 : 5B
AEE0 0F EE 0E DE 1D 2E 1C 9A : EA
AEE8 1A 1C 19 B4 17 5E 16 5E : EC
AEF0 16 1E 15 EE 13 D0 12 00 : 2C
AEF8 03 80 01 C0 00 80 00 60 : 24
AF00 00 4D 00 40 00 37 00 30 : F4
AF08 00 2A 00 26 00 23 00 20 : 93
AF10 00 1D 00 1B 00 1A 00 18 : 6A
AF18 00 16 00 15 00 14 00 13 : 52
AF20 00 12 00 11 00 10 00 10 : 43
AF28 00 0F 00 0F 00 0E 00 0E : 3A
---------------------------------
SUM: 3A C3 A8 91 3D 22 EB 43 FEE9

AF30 00 0D 00 0D 00 0C 00 0C : 32
AF38 00 00 00 00 00 00 00 00 : 00
AF40 00 00 00 00 00 00 00 00 : 00
AF48 00 04 64 08 04 04 64 08 : E4
AF50 04 04 64 08 04 04 64 08 : E8
AF58 04 04 64 08 04 04 64 08 : E8
AF60 04 04 64 08 04 04 64 08 : E8
AF68 04 04 64 08 04 04 64 08 : E8
AF70 04 04 64 08 04 04 64 08 : E8
AF78 04 04 64 08 04 04 64 08 : E8
AF80 04 04 64 08 04 04 64 08 : E8
AF88 04 04 64 08 04 00 00 00 : 78
AF90 00 00 00 00 00 00 00 00 : 00
AF98 00 00 00 00 00 00 00 00 : 00
AFA0 00 00 00 00 00 00 00 00 : 00
AFA8 00 00 00 00 00 00 00 00 : 00
---------------------------------
SUM: 20 31 84 55 24 2C 20 4C 0055

AFB0 00 00 00 00 00 00 00 00 : 00
AFB8 00 00 00 00 00 0A 00 0A : 14
AFC0 14 1E 28 32 3C 40 50 55 : AD
AFC8 5F 64 6E 78 7F 00 00 00 : 28
AFD0 00 00 00 00 00 00 00 00 : 00
AFD8 00 00 00 00 00 00 00 00 : 00
AFE0 00 00 00 00 00 00 00 00 : 00
AFE8 00 00 00 00 00 00 00 00 : 00
AFF0 00 00 00 00 00 00 00 00 : 00
AFF8 00 00 00 00 00 00 00 00 : 00
B000 D5 CD FA AD DA DA 3D 7D : B7
B008 B7 28 0B 3D FE E4 D2 DA : B5
B010 3D FE 28 D2 2A 3F F5 3E : D1
B018 01 CD E3 3D 7B FE 08 30 : 9F
B020 43 F1 D1 D5 21 39 40 19 : 8D
B028 77 6F CD 5B 3E D1 C5 E5 : C7
---------------------------------
SUM: F7 A2 44 D3 97 4F 61 22 FCE1

B030 3E 20 83 CD DE 3D C6 10 : 9F
B038 16 14 CD 3F AE C6 08 CD : 7F
B040 DE 3D 1E 06 CD 6A 3E 1D : D1
B048 20 FA 11 05 00 19 3E 18 : 9F
B050 CD DE 3D 3C CD DE 3D CD : D9
B058 DE 3D 3E 1B CD DE 3D E1 : 3D
B060 C1 C3 68 AB F1 CD CC AE : CF
B068 23 3E 0C 77 D1 C3 68 AB : 8B
B070 56 23 C3 3F AE E5 21 ED : 1C
B078 AF 19 77 E1 C9 F5 D5 CD : 80
B080 CC AE 23 7E F5 21 39 40 : AA
B088 19 6E CD 5B 3E 7E E6 07 : 58
B090 0E 07 FE 04 38 0B CB 89 : AE
B098 28 07 CB 91 FE 07 20 01 : B1
B0A0 0D 11 06 00 19 11 81 40 : 0F
B0A8 06 04 7E CB 09 30 02 F6 : 84
---------------------------------
SUM: 14 02 E5 E9 B7 9E 7B DA 6CE5

B0B0 80 12 23 13 10 F4 06 04 : D6
B0B8 EB 2B 3E 7F BE 38 01 7E : 48
B0C0 2B 10 F9 C1 ED 44 90 C6 : 7C
B0C8 7F 4F D1 D5 3E 58 83 F5 : 82
B0D0 23 E5 06 04 7E B7 FA 4D : 8E
B0D8 3E 81 F2 4D 3E 3E 7F E6 : DF
B0E0 7F 77 23 10 EF E1 F1 CD : B7
B0E8 6A 3E D1 F1 C9 26 00 29 : 82
B0F0 29 54 5D 29 29 29 19 11 : 7F
B0F8 90 B1 19 C9 C6 08 CD DE : 9C
B100 3D C6 10 CD DE 3D D6 08 : D9
B108 CD DE 3D C6 10 CD DE 3D : A6
B110 C9 F5 C5 E5 3E 09 01 01 : B1
B118 00 21 94 3E 04 ED A3 3D : C4
B120 20 FA E1 C1 F1 C9 18 01 : 8F
B128 00 03 C1 04 44 05 68 E6 : 5F
---------------------------------
SUM: 0B 73 D5 E7 C1 C3 42 BF 8373

B130 7F C5 F5 01 01 00 ED 78 : A0
B138 CB 57 28 FA F1 0B ED 79 : A6
B140 C1 C9 F5 CD 1A 40 20 11 : D7
B148 F1 CD E3 3E 77 CD D5 3E : 36
B150 CD CC AE 23 7E CD 9F 3E : 92
B158 C9 F1 C9 CD E3 3E 7E CD : BC
B160 D5 3E AF CD 9F 3E C9 F5 : 2A
B168 CD 22 3F C6 90 CD 9F 3E : 2E
B170 F1 CD 9F 3E C9 16 00 21 : 9B
B178 DD AF 19 C9 7D D5 F5 CD : 82
B180 CC AE 7E 3C 07 07 5F 07 : A8
B188 83 5F F1 FE 05 30 02 C6 : CE
B190 0E FE 0B 38 01 3D D6 05 : 68
B198 83 D1 C9 CD FA AD 38 11 : DA
B1A0 7D FE 11 30 0C B7 28 09 : B0
B1A8 3D 16 00 21 29 40 19 77 : 6D
---------------------------------
SUM: 9C 3B 66 20 95 31 F9 CF 0A76

B1B0 3C C3 68 AB E5 21 29 40 : 81
B1B8 19 7E E1 C9 D1 FE 64 DA : 4E
B1C0 68 AB D6 64 6F CD 22 3F : EA
B1C8 C6 C0 CD 9F 3E 7D CD 9D : 17
B1D0 3E AF CD E3 3D 3C C3 68 : 41
B1D8 AB ED 78 FE 28 28 09 FE : 65
B1E0 5B 20 1E 3E F0 CD 9F 3E : 71
B1E8 03 CD FA AD 38 0A 7D CD : 03
B1F0 9F 3E ED 78 FE 2C 28 F0 : 84
B1F8 FE 5D 20 05 3E F7 CD 9F : 21
B200 3E 03 C3 68 AB CD FA AD : 8B
B208 38 21 ED 78 FE 2C 20 1B : 23
B210 E5 03 CD FA AD 7D E1 38 : F2
B218 12 67 CD 22 3F C6 B0 CD : EA
B220 9F 3E 7D CD 9D 3E 7C CD : 4B
B228 9D 3E 3C C3 68 AB CD FA : B4
---------------------------------
SUM: 10 DA 59 4C C6 EC 4D 8A 3560

B230 AD 38 19 AF CB 1D CB 1F : 7F
B238 CB 1F F5 CD 22 3F C6 E0 : B3
B240 CD 9F 3E F1 CD 9D 3E 7D : C0
B248 CD 9D 3E 3C C3 68 AB CD : 87
B250 FA AD 38 17 7D 32 D8 AE : 2B
B258 ED 78 FE 2C 20 0D 03 CD : 8C
B260 FA AD 38 07 7D 32 D7 AE : 1A
B268 CD 2E AA C3 68 AB CD FA : 42
B270 AD 7D B7 28 1B FE 10 30 : 62
B278 18 F5 CD 22 3F C6 B0 CD : 7E
B280 9F 3E 21 BD AF 7E CD 9D : 52
B288 3E F1 85 6F 7E CD 9D 3E : 49
B290 3C C3 68 AB E5 21 ED AF : B4
B298 19 7E E1 B7 C9 AF D5 E5 : 61
B2A0 ED 5B FD AF 21 CD AF 19 : AA
B2A8 77 E1 D1 C9 D5 E5 ED 5B : F4
---------------------------------
SUM: 1B B1 E3 06 2A 0E 81 4C 9E25

B2B0 FD AF 21 CD AF 19 7E E1 : C1
B2B8 D1 B7 C9 00 01 02 03 04 : 5B
B2C0 05 06 07 08 09 0A 0B 0C : 44
B2C8 0D 0E 0F 01 01 01 01 01 : 2F
B2D0 01 01 01 00 00 00 00 00 : 03
B2D8 00 00 00 00 00 00 00 00 : 00
B2E0 00 00 00 00 00 00 00 00 : 00
B2E8 00 00 00 00 00 00 00 00 : 00
B2F0 00 00 00 00 00 00 00 00 : 00
B2F8 00 00 00 00 00 00 00 00 : 00
B300 00 00 00 00 00 00 00 00 : 00
B308 00 00 00 00 00 00 00 00 : 00
B310 00 00 00 00 00 00 00 E5 : E5
B318 21 98 40 35 20 0A F5 3E : 8B
B320 F8 CD 9F 3E 3E 04 77 F1 : 4C
B328 E1 C9 04 00 00 00 00 00 : AE
---------------------------------
SUM: DB A9 E4 49 18 34 F9 06 FA57

B330 00 00 00 00 00 00 00 00 : 00
B338 00 00 00 00 00 00 00 00 : 00
B340 00 00 00 00 00 00 00 00 : 00
B348 00 00 00 00 00 00 00 00 : 00
B350 00 00 00 00 00 00 00 00 : 00
B358 00 00 00 00 00 00 00 00 : 00
B360 00 00 00 00 00 00 00 00 : 00
B368 00 00 00 00 00 00 00 00 : 00
B370 00 00 00 00 00 00 00 00 : 00
B378 00 00                   : 00
---------------------------------
SUM: 00 00 00 00 00 00 00 00 0000
```

リスト4 MIDIシーケンサ(ソースリスト)

```
0000                          1 ;         Iwai's MML with MIDI
0000                          2 ;
0000                          3 ;         Power uped by S.Kaneko
0000                          4 ;
0000                          5 ;          255PS (5500rpm.NET)
0000                          6                 ;
A8B0                          7                 ORG      $A8B0
A8B0                          8                 ;
A8B0                          9                 OFFSET   $C8B0-$A8B0
A8B0                         10                 ;
A8B0                         11 CTC      EQU       $0704
A8B0                         12 WTOP     EQU       $4000
A8B0                         13                 ;
A8B0 21 3E A9                14                 LD       HL,MMLSTART
A8B3 22 E3 2D                15                 LD       ($2DE3),HL
A8B6 22 11 2E                16                 LD       ($2E11),HL
A8B9 22 13 2E                17                 LD       ($2E13),HL
A8BC 21 00 B0                18                 LD       HL,$B000
A8BF 11 6E 3D                19                 LD       DE,$3D6E
A8C2 01 7A 03                20                 LD       BC,$40E8-$3D6E
A8C5 ED B0                   21                 LDIR
A8C7 C9                      22                 RET
A8C8                         23 SPACE
A8C8 00 00 00 00 00 00 00    24                 DS       $A900-SPACE  ; ORG $A900
A8CF 00 00 00 00 00 00 00
A8D6 00 00 00 00 00 00 00
A8DD 00 00 00 00 00 00 00
A8E4 00 00 00 00 00 00 00
A8EB 00 00 00 00 00 00 00
A8F2 00 00 00 00 00 00 00
A8F9 00 00 00 00 00 00 00
A900                         25                 ;
A900                         26 STACK
A900 00 00 00 00 00 00 00    27                 DS       30*2
A907 00 00 00 00 00 00 00
A90E 00 00 00 00 00 00 00
A915 00 00 00 00 00 00 00
A91C 00 00 00 00 00 00 00
A923 00 00 00 00 00 00 00
A92A 00 00 00 00 00 00 00
A931 00 00 00 00 00 00 00
A938 00 00 00 00
A93C                         28 INTSP
A93C 00 00                   29                 DS       2
A93E                         30 MMLSTART
A93E CD D1 7F                31                 CALL     $7FD1
A941 3A DB A5                32                 LD       A,($A5DB)
A944 E5                      33                 PUSH     HL
A945 EB                      34                 EX       DE,HL
A946 FE 03                   35                 CP       03
A948 CA D4 A9                36                 JP       Z,DOSTR
A94B                         37 ;
A94B                         38 ;<<<<<<<<<<<<<< TEMPO >>>>>>>>>>>>>>>>>>
A94B                         39 STEMPO
A94B CD 36 54                40                 CALL     $5436
A94E CD AF 5B                41                 CALL     $5BAF
A951 7C                      42                 LD       A,H
A952 B5                      43                 OR       L
A953 20 4E                   44                 JR       NZ,PRET
A955                         45 INIT
A955 CD 7F 3E                46                 CALL     INISIO
A958 21 01 40                47                 LD       HL,WTOP+1
A95B 22 2A AA                48                 LD       (HEAD),HL
A95E 22 2C AA                49                 LD       (HEAD0),HL
A961 2B                      50                 DEC      HL
A962 22 28 AA                51                 LD       (TAIL),HL
A965 44 4D                   52                 LD       BC,HL
A967 AF                      53                 XOR      A
A968 ED 79                   54                 OUT      (C),A
A96A 3E 1E                   55                 LD       A,30     ; T120
A96C 32 D8 AE                56                 LD       (TMPV),A
A96F 3E 2B                   57                 LD       A,43     ; for 384
A971 32 D7 AE                58                 LD       (TMPV0),A
A974 32 FF AF                59                 LD       (PFLAG),A
A977                         60                 ;
A977 21 49 AF                61                 LD       HL,DFLDMY
A97A 11 4D AF                62                 LD       DE,DFLW
A97D 01 40 00                63                 LD       BC,4*16
A980 ED B0                   64                 LDIR
A982                         65                 ;
A982 EB                      66                 EX       DE,HL
A983 06 10                   67                 LD       B,16
A985                         68 INIT0
A985 36 00                   69                 LD       (HL),0
A987 23                      70                 INC      HL
A988 10 FB                   71                 DJNZ     INIT0
A98A                         72                 ;
A98A F3                      73                 DI
A98B 3E C3                   74                 LD       A,$C3
A98D 32 3C 01                75                 LD       ($013C),A
A990 21 A5 A9                76                 LD       HL,QUIET1
A993 22 3D 01                77                 LD       ($013D),HL
A996 CD 3C 01                78                 CALL     $013C
A999                         79                 ;
A999 CD 2E AA                80                 CALL     TOCTC
A99C 21 4B AA                81                 LD       HL,IEXEC
A99F 22 5E 00                82                 LD       ($005E),HL
A9A2 FB                      83                 EI
A9A3                         84 PRET
A9A3 E1                      85                 POP      HL
A9A4 C9                      86                 RET
A9A5                         87 QUIET1
A9A5 01 07 07                88                 LD       BC,CTC+3
A9A8 3E 03                   89                 LD       A,3
A9AA ED 79                   90                 OUT      (C),A
A9AC                         91                 ;
A9AC 16 07                   92                 LD       D,7
A9AE 3E 08                   93                 LD       A,8
A9B0                         94 Q2
A9B0 CD 3F AE                95                 CALL     WOPM
A9B3 15                      96                 DEC      D
A9B4 20 FA                   97                 JR       NZ,Q2
A9B6 CD 3F AE                98                 CALL     WOPM
A9B9                         99                 ;
A9B9 3E B0                  100                 LD       A,$B0
A9BB                        101 Q3
A9BB F5                     102                 PUSH     AF
A9BC CD 9F 3E               103                 CALL     MIDI   ; Bn,123,0
A9BF 3E 7B                  104                 LD       A,123 ;(All Note off)
A9C1 CD 9F 3E               105                 CALL     MIDI
A9C4 AF                     106                 XOR      A
A9C5 CD 9F 3E               107                 CALL     MIDI
A9C8 F1                     108                 POP      AF
A9C9 3C                     109                 INC      A
A9CA FE C0                  110                 CP       $C0
A9CC 20 ED                  111                 JR       NZ,Q3
A9CE                        112                 ;
A9CE 01 00 1C               113                 LD       BC,$1C00
A9D1 C3 3F 01               114                 JP       $013F
A9D4                        115 ;
A9D4                        116 ;<<<<<<<<<<<<<< DO STRING >>>>>>>>>>>>>>
A9D4                        117 ;
A9D4                        118 DOSTR
A9D4 CD C3 7F               119                 CALL     $7FC3
A9D7 B7                     120                 OR       A
A9D8 28 03                  121                 JR       Z,PRETX
A9DA CD F5 A9               122                 CALL     STORE
A9DD                        123 PRETX
A9DD E1                     124                 POP      HL
A9DE 7E                     125                 LD       A,(HL)
A9DF D6 3B                  126                 SUB      ';'
A9E1 23                     127                 INC      HL
A9E2 C8                     128                 RET      Z
A9E3 2B                     129                 DEC      HL
A9E4                        130 PRETXX
A9E4 ED 4B 2C AA            131                 LD       BC,(HEAD0)
A9E8 AF                     132                 XOR      A
A9E9 ED 79                  133                 OUT      (C),A
A9EB 03                     134                 INC      BC
A9EC ED 43 2C AA            135                 LD       (HEAD0),BC
A9F0 ED 43 2A AA            136                 LD       (HEAD),BC
A9F4 C9                     137                 RET
A9F5                        138 STORE
A9F5 ED 4B 2C AA            139                 LD       BC,(HEAD0)
A9F9 2A 2A AA               140                 LD       HL,(HEAD)
A9FC B7                     141                 OR       A
A9FD ED 42                  142                 SBC      HL,BC
A9FF 20 05                  143                 JR       NZ,STORE1
AA01 2E 3A                  144                 LD       L,':'
AA03 ED 69                  145                 OUT      (C),L
AA05 03                     146                 INC      BC
AA06                        147 STORE1
AA06 6F                     148                 LD       L,A
AA07                        149 SLOOP
AA07 1A                     150                 LD       A,(DE)
AA08 FE 61                  151                 CP       'a'
AA0A 38 06                  152                 JR       C,S0
AA0C FE 7B                  153                 CP       'z'+1
AA0E 30 02                  154                 JR       NC,S0
AA10 C6 E0                  155                 ADD      A,'A'-'a'
AA12                        156 S0
AA12 ED 79                  157                 OUT      (C),A
AA14 03                     158                 INC      BC
AA15 13                     159                 INC      DE
AA16 78                     160                 LD       A,B
AA17 B1                     161                 OR       C
AA18 28 08                  162                 JR       Z,FULL
AA1A                        163                 ;
AA1A 2D                     164                 DEC      L
AA1B 20 EA                  165                 JR       NZ,SLOOP
AA1D                        166                 ;
AA1D ED 43 2C AA            167                 LD       (HEAD0),BC
AA21 C9                     168                 RET
AA22                        169 FULL
AA22 CD 3C 01               170                 CALL     $013C
AA25 C3 27 20               171                 JP       $2027
AA28                        172                 ;
AA28 00 40                  173 TAIL     DEFW     $4000
AA2A 00 40                  174 HEAD     DEFW     $4000
AA2C 00 40                  175 HEAD0    DEFW     $4000
AA2E                        176                 ;
AA2E                        177 TOCTC
AA2E 01 04 07               178                 LD       BC,CTC
AA31 3E 07                  179                 LD       A,7
AA33 ED 79                  180                 OUT      (C),A
AA35 3A D7 AE               181                 LD       A,(TMPV0)
AA38 ED 79                  182                 OUT      (C),A
AA3A 3E 58                  183                 LD       A,$58    ; SET V
AA3C ED 79                  184                 OUT      (C),A
AA3E                        185 TOCTC1
AA3E 01 07 07               186                 LD       BC,CTC+3
AA41 3E C7                  187                 LD       A,$C7
AA43 ED 79                  188                 OUT      (C),A
AA45 3A D8 AE               189                 LD       A,(TMPV)
AA48 ED 79                  190                 OUT      (C),A
AA4A C9                     191                 RET
AA4B                        192 ;
AA4B                        193 ;<<<<<<<<<<<<<< INT ROUTINE >>>>>>>>>>>>
AA4B                        194 ;
AA4B                        195 IEXEC
AA4B F3                     196                 DI
AA4C ED 73 3C A9            197                 LD       (INTSP),SP
AA50 31 3C A9               198                 LD       SP,INTSP
AA53 F5                     199                 PUSH     AF
AA54 C5                     200                 PUSH     BC
AA55 D5                     201                 PUSH     DE
AA56 E5                     202                 PUSH     HL
AA57                        203                 ;
AA57 21 FF AF               204                 LD       HL,PFLAG
AA5A 7E                     205                 LD       A,(HL)
AA5B B7                     206                 OR       A
AA5C CA D3 AA               207                 JP       Z,IE9 ; 0 = PAUSE
AA5F                        208                 ;
AA5F 00 00 00               209                 DS       3 ; for Timing Clock
AA62                        210                 ;
AA62 21 8D AF               211                 LD       HL,ACTF
AA65 06 10                  212                 LD       B,16
AA67 AF                     213                 XOR      A
AA68                        214 IE0
AA68 B6                     215                 OR       (HL)
AA69 23                     216                 INC      HL
AA6A 20 51                  217                 JR       NZ,IE1
AA6C 10 FA                  218                 DJNZ     IE0
AA6E                        219                 ;
AA6E ED 4B 28 AA            220                 LD       BC,(TAIL)
AA72                        221 IEQ
AA72 2A 2A AA               222                 LD       HL,(HEAD)
AA75 2B                     223                 DEC      HL
AA76 B7                     224                 OR       A
AA77 ED 42                  225                 SBC      HL,BC
AA79 20 07                  226                 JR       NZ,IEQ2
AA7B 3E 01                  227                 LD       A,1
AA7D 32 FF AF               228                 LD       (PFLAG),A
AA80 18 51                  229                 JR       IE9
AA82                        230 IEQ2
AA82 ED 78                  231                 IN       A,(C)
AA84 03                     232                 INC      BC
AA85 28 0C                  233                 JR       Z,START
AA87                        234 DROP
AA87 ED 78                  235                 IN       A,(C)
AA89 03                     236                 INC      BC
AA8A 20 FB                  237                 JR       NZ,DROP
AA8C 0B                     238                 DEC      BC
AA8D ED 43 28 AA            239                 LD       (TAIL),BC
AA91 18 DF                  240                 JR       IEQ
AA93                        241 START
AA93 ED 43 28 AA            242                 LD       (TAIL),BC
AA97 21 8D AF               243                 LD       HL,ACTF
AA9A E5                     244                 PUSH     HL
AA9B 21 9D AF               245                 LD       HL,PC
AA9E 16 00                  246                 LD       D,0
AAA0                        247 SPC0
AAA0 ED 78                  248                 IN       A,(C)
AAA2 28 05                  249                 JR       Z,SPC1
AAA4 03                     250                 INC      BC
AAA5 FE 3A                  251                 CP       ':'
AAA7 20 F7                  252                 JR       NZ,SPC0
AAA9                        253 SPC1
AAA9 B7                     254                 OR       A
AAAA 28 02                  255                 JR       Z,SPC2
AAAC 3E 01                  256                 LD       A,1
```

```
AAAE                       257 SPC2
AAAE E3                    258          EX      (SP),HL
AAAF 77                    259          LD      (HL),A
AAB0 23                    260          INC     HL
AAB1 E3                    261          EX      (SP),HL
AAB2 71                    262          LD      (HL),C
AAB3 23                    263          INC     HL
AAB4 70                    264          LD      (HL),B
AAB5 23                    265          INC     HL
AAB6 14                    266          INC     D
AAB7 7A                    267          LD      A,D
AAB8 FE 10                 268          CP      16
AABA 38 E4                 269          JR      C,SPC0
AABC E1                    270          POP     HL
AABD                       271 IE1
AABD 11 00 00              272          LD      DE,0
AAC0 21 8D AF              273          LD      HL,ACTF
AAC3                       274 IEL
AAC3 D5                    275          PUSH    DE
AAC4 E5                    276          PUSH    HL
AAC5 7E                    277          LD      A,(HL)
AAC6 B7                    278          OR      A
AAC7 C4 DE AA              279          CALL    NZ,ACT
AACA E1                    280          POP     HL
AACB D1                    281          POP     DE
AACC 23                    282          INC     HL
AACD 1C                    283          INC     E
AACE 7B                    284          LD      A,E
AACF FE 10                 285          CP      16
AAD1 20 F0                 286          JR      NZ,IEL
AAD3                       287 IE9
AAD3 E1                    288          POP     HL
AAD4 D1                    289          POP     DE
AAD5 C1                    290          POP     BC
AAD6 F1                    291          POP     AF
AAD7 ED 7B 3C A9           292          LD      SP,(INTSP)
AADB FB                    293          EI
AADC ED 4D                 294          RETI
AADE                       295 ;
AADE                       296 ;<<<<<<<<<<<<<< SUB ROUTINE >>>>>>>>>>>>
AADE                       297 ;
AADE                       298 ACT
AADE E5                    299          PUSH    HL
AADF D5                    300          PUSH    DE
AAE0 F2 F5 AA              301          JP      P,ACT7
AAE3                       302          ;
AAE3 CD 4A AE              303          CALL    RCTR
AAE6 78                    304          LD      A,B
AAE7 B1                    305          OR      C
AAE8 D1                    306          POP     DE
AAE9 E5                    307          PUSH    HL
AAEA CC 36 AD              308          CALL    Z,KEYOFF
AAED E1                    309          POP     HL
AAEE 7C                    310          LD      A,H
AAEF B5                    311          OR      L
AAF0 E1                    312          POP     HL
AAF1 C0                    313          RET     NZ
AAF2 36 01                 314          LD      (HL),1
AAF4 C9                    315          RET
AAF5                       316 ACT7
AAF5 CB 23                 317          SLA     E
AAF7 21 9D AF              318          LD      HL,PC
AAFA 19                    319          ADD     HL,DE
AAFB 4E                    320          LD      C,(HL)
AAFC 23                    321          INC     HL
AAFD 46                    322          LD      B,(HL)
AAFE                       323          ;
AAFE ED 78                 324          IN      A,(C)
AB00 28 04                 325          JR      Z,DIE
AB02 FE 3A                 326          CP      ':'
AB04 20 08                 327          JR      NZ,WAKE
AB06                       328 DIE
AB06 D1                    329          POP     DE
AB07                       330 DIE2
AB07 70                    331          LD      (HL),B
AB08 2B                    332          DEC     HL
AB09 71                    333          LD      (HL),C
AB0A                       334          ;
AB0A E1                    335          POP     HL
AB0B 36 00                 336          LD      (HL),0
AB0D C9                    337          RET
AB0E                       338 WAKE
AB0E D1                    339          POP     DE
AB0F                       340 WUL
AB0F ED 78                 341          IN      A,(C)
AB11 28 F4                 342          JR      Z,DIE2
AB13 FE 3A                 343          CP      ':'
AB15 28 F0                 344          JR      Z,DIE2
AB17 D5                    345          PUSH    DE
AB18 E5                    346          PUSH    HL
AB19 CD 29 AB              347          CALL    DONOTE
AB1C E1                    348          POP     HL
AB1D 28 E7                 349          JR      Z,DIE
AB1F D1                    350          POP     DE
AB20 38 ED                 351          JR      C,WUL
AB22                       352          ;
AB22 70                    353          LD      (HL),B
AB23 2B                    354          DEC     HL
AB24 71                    355          LD      (HL),C
AB25                       356          ;
AB25 E1                    357          POP     HL
AB26 36 80                 358          LD      (HL),$80
AB28 C9                    359          RET
AB29                       360 ;
AB29                       361 ;<<<<<<<<<<<<<< DO NOTE >>>>>>>>>>>>>>>>
AB29                       362 ;
AB29                       363 DONOTE
AB29 ED 53 FD AF           364          LD      (DESTACK),DE
AB2D ED 78                 365          IN      A,(C)
AB2F 03                    366          INC     BC
AB30 FE 41                 367          CP      'A'
AB32 38 2A                 368          JR      C,DN0
AB34 FE 48                 369          CP      'G'+1
AB36 30 34                 370          JR      NC,DN1
AB38                       371          ;
AB38 CD 49 AB              372          CALL    GETNT
AB3B                       373 RN0
AB3B C5                    374          PUSH    BC
AB3C D5                    375          PUSH    DE
AB3D CD B1 AC              376          CALL    KEYON
AB40 D1                    377          POP     DE
AB41 C1                    378          POP     BC
AB42                       379 RN1
AB42 CD 4E AD              380          CALL    DONUM
AB45                       381 RN2
AB45 3E 01                 382          LD      A,1
AB47 B7                    383          OR      A
AB48 C9                    384          RET
AB49                       385 GETNT
AB49 D6 41                 386          SUB     'A'     ; A- = 0
AB4B 87                    387          ADD     A,A     ; A  = 1
AB4C 6F                    388          LD      L,A     ; A+ = 2
AB4D ED 78                 389          IN      A,(C)   ; B  = 3
AB4F 03                    390          INC     BC      ; B+ = 4
AB50 FE 2D                 391          CP      '-'     ; C  = 5
AB52 C8                    392          RET     Z       ; C+ = 6
AB53 2C                    393          INC     L       ; D  = 7
```

```
AB54 2C                    394          INC     L       ; D+ = 8
AB55 FE 2B                 395          CP      '+'     ; E  = 9
AB57 C8                    396          RET     Z       ; E+ =10
AB58 FE 23                 397          CP      '#'     ; F  =11
AB5A C8                    398          RET     Z       ; F+ =12
AB5B 2D                    399          DEC     L       ; G  =13
AB5C 0B                    400          DEC     BC      ; G+ =14
AB5D C9                    401          RET
AB5E                       402 DN0
AB5E FE 3E                 403          CP      '>'
AB60 CA 6B AC              404          JP      Z,UPDOWN
AB63 FE 3C                 405          CP      '<'
AB65 CA 6B AC              406          JP      Z,UPDOWN
AB68                       407 ACT8
AB68 7F                    408          LD      A,A
AB69 B7                    409          OR      A
AB6A 37                    410          SCF
AB6B C9                    411          RET
AB6C                       412          ;
AB6C                       413 DN1
AB6C FE 52 28 D2           414          CP      'R'     :JR     Z,RN1
AB70 FE 49 CA 6E 3D        415          CP      'I'     :JP     Z,INST00
AB75 FE 5A CA 47 3F        416          CP      'Z'     :JP     Z,TARE
AB7A FE 5E CA 73 3F        417          CP      '^'     :JP     Z,CTRLC
AB7F FE 5C CA 9C 3F        418          CP      '¥'     :JP     Z,BEND
AB84 FE 4D CA 09 3F        419          CP      'M'     :JP     Z,MIDNUM
AB89 FE 54 CA 7D AC        420          CP      'T'     :JP     Z,STRUN
AB8E FE 59 CA 3D AC        421          CP      'Y'     :JP     Z,WREG
AB93 FE 4B CA 1D AC        422          CP      'K'     :JP     Z,KEYFR
AB98 FE 7E CA 98 AE        423          CP      '~'     :JP     Z,VUP
AB9D FE 5F CA 78 AE        424          CP      '_'     :JP     Z,VDOWN
ABA2 FE 53 CA BD 3F        425          CP      'S'     :JP     Z,STCTC
ABA7 FE 50 28 39           426          CP      'P'     :JR     Z,PAN
ABAB 21 00 00              427          LD      HL,$0000
ABAE FE 4F 28 0F           428          CP      'O'     :JR     Z,SETGO
ABB2 2C                    429          INC     L
ABB3 FE 56 28 0A           430          CP      'V'     :JR     Z,SETGO
ABB7 2C                    431          INC     L
ABB8 FE 51 28 05           432          CP      'Q'     :JR     Z,SETGO
ABBC 2C                    433          INC     L
ABBD FE 4C 20 A7           434          CP      'L'     :JR     NZ,ACT8
ABC1                       435 SETGO
ABC1 E5                    436          PUSH    HL
ABC2 CD CC AE              437          CALL    DFLTBL
ABC5 D1                    438          POP     DE
ABC6 19                    439          ADD     HL,DE
ABC7 E5                    440          PUSH    HL
ABC8 F5                    441          PUSH    AF
ABC9 CD FA AD              442          CALL    NUMBER
ABCC 30 04                 443          JR      NC,STG1
ABCE F1                    444          POP     AF
ABCF D1                    445          POP     DE
ABD0 18 96                 446          JR      ACT8
ABD2                       447 STG1
ABD2 67                    448          LD      H,A
ABD3 F1                    449          POP     AF
ABD4 D1                    450          POP     DE
ABD5 EB                    451          EX      DE,HL
ABD6 73                    452          LD      (HL),E
ABD7 FE 4C                 453          CP      'L'
ABD9 20 8D                 454          JR      NZ,ACT8
ABDB 7A                    455          LD      A,D
ABDC FE 02                 456          CP      '0'-'.'
ABDE 20 88                 457          JR      NZ,ACT8
ABE0 CB FE                 458          SET     7,(HL)
ABE2 18 84                 459          JR      ACT8
ABE4                       460 PAN
ABE4 CD 02 40              461          CALL    MDCHECK
ABE7 CA DC 3F              462          JP      Z,MIDIPAN
ABEA                       463          ;
ABEA 7B                    464          LD      A,E
ABEB FE 08                 465          CP      8
ABED D2 68 AB              466          JP      NC,ACT8
ABF0 C5                    467          PUSH    BC
ABF1 D5                    468          PUSH    DE
ABF2 CD FA AD              469          CALL    NUMBER
ABF5 38 21                 470          JR      C,PAN2
ABF7 7D                    471          LD      A,L
ABF8 FE 04                 472          CP      4
ABFA 30 1C                 473          JR      NC,PAN2
ABFC D1                    474          POP     DE
ABFD D5                    475          PUSH    DE
ABFE D5                    476          PUSH    DE
ABFF F5                    477          PUSH    AF
AC00 21 39 40              478          LD      HL,INSTN
AC03 19                    479          ADD     HL,DE
AC04 6E                    480          LD      L,(HL)
AC05 CD 5B 3E              481          CALL    GETVTD
AC08 7E                    482          LD      A,(HL)
AC09 E6 3F                 483          AND     $3F
AC0B 4F                    484          LD      C,A
AC0C F1                    485          POP     AF
AC0D 0F                    486          RRCA
AC0E 0F                    487          RRCA
AC0F B1                    488          OR      C
AC10 D1                    489          POP     DE
AC11 57                    490          LD      D,A
AC12 3E 20                 491          LD      A,$20
AC14 83                    492          ADD     A,E
AC15 CD 3F AE              493          CALL    WOPM
AC18                       494 PAN2
AC18 D1                    495          POP     DE
AC19 C1                    496          POP     BC
AC1A C3 68 AB              497          JP      ACT8
AC1D                       498 KEYFR
AC1D 7B                    499          LD      A,E
AC1E FE 08                 500          CP      8
AC20 D2 68 AB              501          JP      NC,ACT8
AC23 C5                    502          PUSH    BC
AC24 D5                    503          PUSH    DE
AC25 CD FA AD              504          CALL    NUMBER
AC28 D1                    505          POP     DE
AC29 38 0E                 506          JR      C,KEYFR2
AC2B 7D                    507          LD      A,L
AC2C FE 40                 508          CP      $40
AC2E 30 09                 509          JR      NC,KEYFR2
AC30 07                    510          RLCA
AC31 07                    511          RLCA
AC32 57                    512          LD      D,A
AC33 3E 30                 513          LD      A,$30
AC35 83                    514          ADD     A,E
AC36 CD 3F AE              515          CALL    WOPM
AC39                       516 KEYFR2
AC39 C1                    517          POP     BC
AC3A C3 68 AB              518          JP      ACT8
AC3D                       519 WREG
AC3D D5                    520          PUSH    DE
AC3E CD FA AD              521          CALL    NUMBER
AC41 38 24                 522          JR      C,WREG9
AC43 ED 78                 523          IN      A,(C)
AC45 FE 2C                 524          CP      ','
AC47 20 1E                 525          JR      NZ,WREG9
AC49 03                    526          INC     BC
AC4A E5                    527          PUSH    HL
AC4B CD FA AD              528          CALL    NUMBER
AC4E 7D                    529          LD      A,L
AC4F E1                    530          POP     HL
```

```
AC50 38 15          531          JR      C,WREG9
AC52 D1             532          POP     DE
AC53                533          ;
AC53 C5             534          PUSH    BC
AC54 57             535          LD      D,A
AC55 7B             536          LD      A,E
AC56 FE 08          537          CP      8
AC58 7D             538          LD      A,L
AC59 38 07          539          JR      C,WROPM
AC5B CD 2D AE       540          CALL    WPSG
AC5E                541 WREG1
AC5E C1             542          POP     BC
AC5F C3 68 AB       543          JP      ACT8
AC62                544 WROPM
AC62 CD 3F AE       545          CALL    WOPM
AC65 18 F7          546          JR      WREG1
AC67                547 WREG9
AC67 D1             548          POP     DE
AC68 C3 68 AB       549          JP      ACT8
AC6B                550 UPDOWN
AC6B D6 3D          551          SUB     '<'+1
AC6D CD CC AE       552          CALL    DFLTBL
AC70 86             553          ADD     A,(HL)
AC71 FA 68 AB       554          JP      M,ACT8
AC74 FE 09          555          CP      9
AC76 D2 68 AB       556          JP      NC,ACT8
AC79 77             557          LD      (HL),A
AC7A C3 68 AB       558          JP      ACT8
AC7D                559 STRUN
AC7D CD FA AD       560          CALL    NUMBER
AC80 DA 68 AB       561          JP      C,ACT8
AC83 7D             562          LD      A,L
AC84 FE 1E          563          CP      30
AC86 DA 68 AB       564          JP      C,ACT8
AC89 C5             565          PUSH    BC
AC8A 4D             566          LD      C,L
AC8B 06 00          567          LD      B,0
AC8D 11 4E 0E       568          LD      DE,3662
AC90 21 00 00       569          LD      HL,0
AC93 3E 10          570          LD      A,16
AC95                571 STR1
AC95 CB 23          572          SLA     E
AC97 CB 12          573          RL      D
AC99 ED 6A          574          ADC     HL,HL
AC9B B7             575          OR      A
AC9C ED 42          576          SBC     HL,BC
AC9E 30 02          577          JR      NC,STR2
ACA0 09             578          ADD     HL,BC
ACA1 1D             579          DEC     E
ACA2                580 STR2
ACA2 1C             581          INC     E
ACA3 3D             582          DEC     A
ACA4 20 EF          583          JR      NZ,STR1
ACA6 7B             584          LD      A,E
ACA7 32 D8 AE       585          LD      (TMPV),A
ACAA CD 3E AA       586          CALL    TOCTC1
ACAD C1             587          POP     BC
ACAE C3 68 AB       588          JP      ACT8
ACB1                589          ;
ACB1                590 KEYON
ACB1 CD 02 40       591          CALL    MDCHECK
ACB4 CA 2F AD       592          JP      Z,MIDIDO
ACB7                593          ;
ACB7 7B             594          LD      A,E
ACB8 FE 08          595          CP      8
ACBA 38 3A          596          JR      C,OPMDO
ACBC                597 PSGDO
ACBC 7D             598          LD      A,L
ACBD D5             599          PUSH    DE
ACBE CD CC AE       600          CALL    DFLTBL
ACC1 46             601          LD      B,(HL)
ACC2 23             602          INC     HL
ACC3 4E             603          LD      C,(HL)
ACC4 87             604          ADD     A,A
ACC5 5F             605          LD      E,A
ACC6 21 D9 AE       606          LD      HL,PSGFREQ
ACC9 19             607          ADD     HL,DE
ACCA 56             608          LD      D,(HL)
ACCB 23             609          INC     HL
ACCC 5E             610          LD      E,(HL)
ACCD                611 PSG1
ACCD CB 3B          612          SRL     E
ACCF CB 1A          613          RR      D
ACD1 10 FA          614          DJNZ    PSG1
ACD3                615          ;
ACD3 E1             616          POP     HL
ACD4 61             617          LD      H,C
ACD5                618          ;
ACD5 7D             619          LD      A,L     ;FREQ
ACD6 D6 08          620          SUB     8
ACD8 87             621          ADD     A,A
ACD9 CD 2D AE       622          CALL    WPSG
ACDC 3C             623          INC     A
ACDD 53             624          LD      D,E
ACDE CD 2D AE       625          CALL    WPSG
ACE1                626          ;
ACE1 7D             627          LD      A,L     ;VOL
ACE2 54             628          LD      D,H
ACE3 CD 2D AE       629          CALL    WPSG
ACE6                630          ;
ACE6 3E 07          631          LD      A,7
ACE8 16 F8          632          LD      D,$F8
ACEA CD 2D AE       633          CALL    WPSG
ACED 3E 0D          634          LD      A,13
ACEF CD 36 AE       635          CALL    RPSG
ACF2 CD 2D AE       636          CALL    WPSG
ACF5 C9             637          RET
ACF6                638 OPMDO
ACF6 7D             639          LD      A,L
ACF7 B7             640          OR      A
ACF8 20 01          641          JR      NZ,OPMDO0
ACFA 3D             642          DEC     A
ACFB                643 OPMDO0
ACFB C6 17          644          ADD     A,$17
ACFD FE 1B          645          CP      $1B
ACFF 38 0B          646          JR      C,OPMDO1
AD01 D6 10          647          SUB     $10
AD03 FE 0F          648          CP      $0F
AD05 38 05          649          JR      C,OPMDO1
AD07 FE 12          650          CP      $12
AD09 28 01          651          JR      Z,OPMDO1
AD0B 3C             652          INC     A
AD0C                653 OPMDO1
AD0C D5             654          PUSH    DE
AD0D CD EB 3D       655          CALL    OPMV0
AD10 F5             656          PUSH    AF
AD11 CD CC AE       657          CALL    DFLTBL
AD14 7E             658          LD      A,(HL)
AD15 3D             659          DEC     A
AD16 87             660          ADD     A,A
AD17 87             661          ADD     A,A
AD18 87             662          ADD     A,A
AD19 87             663          ADD     A,A
AD1A 57             664          LD      D,A
AD1B F1             665          POP     AF
AD1C 82             666          ADD     A,D
AD1D 57             667          LD      D,A
AD1E E1             668          POP     HL
AD1F 7D             669          LD      A,L
AD20 C6 28          670          ADD     A,$28
AD22 CD 3F AE       671          CALL    WOPM
AD25 7D             672          LD      A,L
AD26 F6 78          673          OR      $78
AD28 57             674          LD      D,A
AD29 3E 08          675          LD      A,8
AD2B CD 3F AE       676          CALL    WOPM
AD2E C9             677          RET
AD2F                678 MIDIDO
AD2F CD EA 3E       679          CALL    NTNUM
AD32 CD B0 3E       680          CALL    NOTEON
AD35 C9             681          RET
AD36                682 KEYOFF
AD36 CD 02 40       683          CALL    MDCHECK
AD39 CA C9 3E       684          JP      Z,NOTEOFF
AD3C                685          ;
AD3C 7B             686          LD      A,E
AD3D FE 08          687          CP      8
AD3F 38 06          688          JR      C,OFFOPM
AD41                689          ;
AD41 16 00          690          LD      D,0
AD43 CD 2D AE       691          CALL    WPSG
AD46 C9             692          RET
AD47                693 OFFOPM
AD47 53             694          LD      D,E
AD48 3E 08          695          LD      A,8
AD4A CD 3F AE       696          CALL    WOPM
AD4D C9             697          RET
AD4E                698          ;
AD4E                699 DONUM
AD4E D5             700          PUSH    DE
AD4F CD 65 AD       701          CALL    DONUMS
AD52 2B             702          DEC     HL
AD53 1B             703          DEC     DE
AD54 ED 53 63 AD    704          LD      (DONUMW),DE
AD58 D1             705          POP     DE
AD59 C5             706          PUSH    BC
AD5A ED 4B 63 AD    707          LD      BC,(DONUMW)
AD5E CD 66 AE       708          CALL    WCTR
AD61 C1             709          POP     BC
AD62 C9             710          RET
AD63                711          ;
AD63                712 DONUMW
AD63 00 00          713          DS      2
AD65                714          ;
AD65                715 DONUMS
AD65 CD CC AE       716          CALL    DFLTBL
AD68 23             717          INC     HL
AD69 23             718          INC     HL
AD6A 56             719          LD      D,(HL)
AD6B 23             720          INC     HL
AD6C 5E             721          LD      E,(HL)
AD6D D5             722          PUSH    DE
AD6E ED 78          723          IN      A,(C)
AD70 FE 40          724          CP      '@'
AD72 20 21          725          JR      NZ,DONOML
AD74 03             726          INC     BC
AD75 11 00 00       727          LD      DE,0
AD78                728 XNUM0
AD78 CD 1A AE       729          CALL    DIGIT
AD7B 30 05          730          JR      NC,XNUMZ
AD7D 11 01 00       731          LD      DE,1
AD80 18 41          732          JR      DONUM1
AD82                733 XNUMZ
AD82 5F             734          LD      E,A
AD83                735 XNUML
AD83 CD 1A AE       736          CALL    DIGIT
AD86 38 3B          737          JR      C,DONUM1
AD88 62             738          LD      H,D
AD89 6B             739          LD      L,E
AD8A 29             740          ADD     HL,HL
AD8B 29             741          ADD     HL,HL
AD8C 19             742          ADD     HL,DE
AD8D 29             743          ADD     HL,HL
AD8E 5F             744          LD      E,A
AD8F 16 00          745          LD      D,0
AD91 19             746          ADD     HL,DE
AD92 EB             747          EX      DE,HL
AD93 18 EE          748          JR      XNUML
AD95                749 DONOML
AD95 CD FA AD       750          CALL    NUMBER
AD98 30 09          751          JR      NC,DONUM0
AD9A 6B             752          LD      L,E
AD9B CB 7D          753          BIT     7,L
AD9D 28 04          754          JR      Z,DONUM0
AD9F CB BD          755          RES     7,L
ADA1 3E FE          756          LD      A,$FE
ADA3                757 DONUM0
ADA3 F5             758          PUSH    AF
ADA4 7D             759          LD      A,L
ADA5 FE 21          760          CP      33
ADA7 38 02          761          JR      C,DNOK
ADA9 2E 20          762          LD      L,32
ADAB                763 DNOK
ADAB F1             764          POP     AF
ADAC 26 00          765          LD      H,0
ADAE CB 25          766          SLA     L
ADB0 11 F7 AE       767          LD      DE,LENTBL
ADB3 19             768          ADD     HL,DE
ADB4 5E             769          LD      E,(HL)
ADB5 23             770          INC     HL
ADB6 56             771          LD      D,(HL)
ADB7                772          ;
ADB7 FE FE          773          CP      $FE
ADB9 20 08          774          JR      NZ,DONUM1
ADBB 62             775          LD      H,D
ADBC 6B             776          LD      L,E
ADBD CB 3A          777          SRL     D
ADBF CB 1B          778          RR      E
ADC1 19             779          ADD     HL,DE
ADC2 EB             780          EX      DE,HL
ADC3                781 DONUM1
ADC3 62 6B          782          LD      HL,DE
ADC5 F1             783          POP     AF
ADC6                784          ;
ADC6 E5             785          PUSH    HL
ADC7 67             786          LD      H,A
ADC8 CD 0B 40       787          CALL    ANDFLAG0  ; No &
ADCB ED 78          788          IN      A,(C)
ADCD FE 26          789          CP      '&'
ADCF 7C             790          LD      A,H
ADD0 E1             791          POP     HL
ADD1 20 13          792          JR      NZ,DONUM2
ADD3                793          ;
ADD3 03             794          INC     BC
ADD4 ED 78          795          IN      A,(C)
ADD6 03             796          INC     BC
ADD7 FE 2B          797          CP      '+'
ADD9 28 1D          798          JR      Z,DONUM4
ADDB 0B             799          DEC     BC
ADDC 21 FF FF       800          LD      HL,$FFFF
ADDF 3E 01          801          LD      A,1
ADE1 CD 0C 40       802          CALL    ANDFLAG   ; &
ADE4 18 12          803          JR      DONUM4
ADE6                804          ;
```

```
ADE6                     805 DONUM2
ADE6 3D                  806              DEC     A
ADE7 28 03               807              JR      Z,DONUM3
ADE9 19                  808              ADD     HL,DE
ADEA 18 FA               809              JR      DONUM2
ADEC                     810              ;
ADEC                     811 DONUM3
ADEC CB 3C               812              SRL     H
ADEE CB 1D               813              RR      L
ADF0 CB 3C               814              SRL     H
ADF2 CB 1D               815              RR      L
ADF4 CB 3C               816              SRL     H
ADF6 CB 1D               817              RR      L          ; HL=HL/8
ADF8                     818              ;
ADF8                     819 DONUM4
ADF8 EB                  820              EX      DE,HL
ADF9 C9                  821              RET
ADFA                     822              ;
ADFA                     823 NUMBER
ADFA ED 78               824              IN      A,(C)
ADFC 28 1A               825              JR      Z,NUM9
ADFE FE 3A               826              CP      ':'
AE00 28 16               827              JR      Z,NUM9
AE02                     828 NUM0
AE02 CD 1A AE            829              CALL    DIGIT
AE05 D8                  830              RET     C
AE06 6F                  831              LD      L,A
AE07                     832 NUML
AE07 CD 1A AE            833              CALL    DIGIT
AE0A 38 0A               834              JR      C,NUM1
AE0C 67                  835              LD      H,A
AE0D 7D                  836              LD      A,L
AE0E 87                  837              ADD     A,A
AE0F 87                  838              ADD     A,A
AE10 85                  839              ADD     A,L
AE11 87                  840              ADD     A,A
AE12 84                  841              ADD     A,H
AE13 6F                  842              LD      L,A
AE14 18 F1               843              JR      NUML
AE16                     844 NUM1
AE16 B7                  845              OR      A
AE17 C9                  846              RET
AE18                     847 NUM9
AE18 37                  848              SCF
AE19 C9                  849              RET
AE1A                     850 DIGIT
AE1A ED 78               851              IN      A,(C)
AE1C FE 3A               852              CP      '9'+1
AE1E 38 02               853              JR      C,DIGIT1
AE20 37                  854              SCF
AE21 C9                  855              RET
AE22                     856 DIGIT1
AE22 D6 30               857              SUB     '0'
AE24 03                  858              INC     BC
AE25 D0                  859              RET     NC
AE26 FE FE               860              CP      '.'-'0'
AE28 28 01               861              JR      Z,DIGIT2
AE2A 0B                  862              DEC     BC
AE2B                     863 DIGIT2
AE2B 37                  864              SCF
AE2C C9                  865              RET
AE2D                     866              ;
AE2D                     867 WPSG
AE2D 01 00 1C            868              LD      BC,$1C00
AE30 ED 79               869              OUT     (C),A
AE32 05                  870              DEC     B
AE33 ED 51               871              OUT     (C),D
AE35 C9                  872              RET
AE36                     873 RPSG
AE36 01 00 1C            874              LD      BC,$1C00
AE39 ED 79               875              OUT     (C),A
AE3B 05                  876              DEC     B
AE3C ED 50               877              IN      D,(C)
AE3E C9                  878              RET
AE3F                     879              ;
AE3F                     880 WOPM
AE3F C5                  881              PUSH    BC
AE40 01 00 07            882              LD      BC,$0700
AE43 ED 79               883              OUT     (C),A
AE45 0C                  884              INC     C
AE46 ED 51               885              OUT     (C),D
AE48 C1                  886              POP     BC
AE49 C9                  887              RET
AE4A                     888 RCTR
AE4A D5                  889              PUSH    DE
AE4B CB 23               890              SLA     E
AE4D CB 23               891              SLA     E
AE4F 21 41 40            892              LD      HL,PACK
AE52 19                  893              ADD     HL,DE
AE53 4E                  894              LD      C,(HL)
AE54 23                  895              INC     HL
AE55 46                  896              LD      B,(HL)
AE56 23                  897              INC     HL
AE57 5E                  898              LD      E,(HL)
AE58 23                  899              INC     HL
AE59 56                  900              LD      D,(HL)
AE5A EB                  901              EX      DE,HL
AE5B 0B                  902              DEC     BC
AE5C 2B                  903              DEC     HL
AE5D D1                  904              POP     DE
AE5E C5                  905              PUSH    BC
AE5F E5                  906              PUSH    HL
AE60 CD 66 AE            907              CALL    WCTR
AE63 E1                  908              POP     HL
AE64 C1                  909              POP     BC
AE65 C9                  910              RET
AE66                     911 WCTR
AE66 CB 23               912              SLA     E
AE68 CB 23               913              SLA     E
AE6A E5                  914              PUSH    HL
AE6B 21 41 40            915              LD      HL,PACK
AE6E 19                  916              ADD     HL,DE
AE6F 71                  917              LD      (HL),C
AE70 23                  918              INC     HL
AE71 70                  919              LD      (HL),B
AE72 23                  920              INC     HL
AE73 C1                  921              POP     BC
AE74 71                  922              LD      (HL),C
AE75 23                  923              INC     HL
AE76 70                  924              LD      (HL),B
AE77 C9                  925              RET
AE78                     926 VDOWN
AE78 CD FA AD            927              CALL    NUMBER
AE7B 38 05               928              JR      C,VDOWN2
AE7D 7D                  929              LD      A,L
AE7E CD C7 AE            930              CALL    DFLTBLV
AE81 77                  931              LD      (HL),A
AE82                     932 VDOWN2
AE82 CD CC AE            933              CALL    DFLTBL
AE85 23                  934              INC     HL
AE86 7E                  935              LD      A,(HL)
AE87 E5                  936              PUSH    HL
AE88 CD C7 AE            937              CALL    DFLTBLV
AE8B 96                  938              SUB     (HL)
AE8C 38 06               939              JR      C,VRETURN
AE8E E1                  940              POP     HL
AE8F 77                  941              LD      (HL),A
AE90 3C                  942              INC     A          ; Z=0
AE91 C3 68 AB            943              JP      ACT8
AE94                     944 VRETURN
AE94 E1                  945              POP     HL
AE95 C3 68 AB            946              JP      ACT8
AE98                     947 VUP
AE98 CD FA AD            948              CALL    NUMBER
AE9B 38 05               949              JR      C,VUP2
AE9D 7D                  950              LD      A,L
AE9E CD C7 AE            951              CALL    DFLTBLV
AEA1 77                  952              LD      (HL),A
AEA2                     953 VUP2
AEA2 CD 02 40            954              CALL    MDCHECK
AEA5 3E 7F               955              LD      A,127
AEA7 28 09               956              JR      Z,VUP3     ; 0=MIDI
AEA9 7B                  957              LD      A,E
AEAA FE 08               958              CP      8
AEAC 3E 7F               959              LD      A,127
AEAE 38 02               960              JR      C,VUP3
AEB0 3E 0F               961              LD      A,15
AEB2                     962 VUP3
AEB2 CD CC AE            963              CALL    DFLTBL
AEB5 23                  964              INC     HL
AEB6 E5                  965              PUSH    HL
AEB7 F5                  966              PUSH    AF
AEB8 7E                  967              LD      A,(HL)
AEB9 CD C7 AE            968              CALL    DFLTBLV
AEBC 86                  969              ADD     A,(HL)
AEBD 5F                  970              LD      E,A
AEBE F1                  971              POP     AF
AEBF BB                  972              CP      E
AEC0 38 D2               973              JR      C,VRETURN
AEC2 E1                  974              POP     HL
AEC3 73                  975              LD      (HL),E
AEC4 C3 68 AB            976              JP      ACT8
AEC7                     977 DFLTBLV
AEC7 21 39 AF            978              LD      HL,VAREA
AECA 19                  979              ADD     HL,DE
AECB C9                  980              RET
AECC                     981 DFLTBL
AECC D5                  982              PUSH    DE
AECD 21 4D AF            983              LD      HL,DFLW
AED0 CB 23               984              SLA     E
AED2 CB 23               985              SLA     E
AED4 19                  986              ADD     HL,DE
AED5 D1                  987              POP     DE
AED6 C9                  988              RET
AED7                     989 ;
AED7                     990 ;<<<<<<<<<<<<<< WORK AREA >>>>>>>>>>>>>>
AED7                     991 ;
AED7                     992 TMPV0
AED7 00                  993              DS      1
AED8                     994 TMPV
AED8 00                  995              DS      1
AED9                     996 PSGFREQ
AED9 D0 12               997              DW      $12D0
AEDB C2 11               998              DW      $11C2
AEDD C4 10               999              DW      $10C4
AEDF D2 0F              1000              DW      $0FD2
AEE1 EE 0E              1001              DW      $0EEE
AEE3 DE 1D              1002              DW      $1DDE
AEE5 2E 1C              1003              DW      $1C2E
AEE7 9A 1A              1004              DW      $1A9A
AEE9 1C 19              1005              DW      $191C
AEEB B4 17              1006              DW      $17B4
AEED 5E 16              1007              DW      $165E
AEEF 5E 16              1008              DW      $165E
AEF1 1E 15              1009              DW      $151E
AEF3 EE 13              1010              DW      $13EE
AEF5 D0 12              1011              DW      $12D0
AEF7                    1012              ;
AEF7                    1013 LENTBL
AEF7 00 03 80 01 C0 00 80  1014           DW      768,384,192,128,96
AEFE 00 60 00
AF01 4D 00 40 00 37 00 30  1015           DW      77,64,55,48,42
AF08 00 2A 00
AF0B 26 00 23 00 20 00 1D  1016           DW      38,35,32,29,27
AF12 00 1B 00
AF15 1A 00 18 00 16 00 15  1017           DW      26,24,22,21,20
AF1C 00 14 00
AF1F 13 00 12 00 11 00 10  1018           DW      19,18,17,16,16
AF26 00 10 00
AF29 0F 00 0F 00 0E 00 0E  1019           DW      15,15,14,14,13
AF30 00 0D 00
AF33 0D 00 0C 00 0C 00  1020              DW      13,12,12
AF39                    1021              ;
AF39                    1022 VAREA
AF39 00 00 00 00 00 00 00  1023           DS      16
AF40 00 00 00 00 00 00 00
AF47 00 00
AF49                    1024 DFLDMY
AF49 04 64 08 04        1025              DB      4,100,8,4
AF4D                    1026 DFLW
AF4D 04 64 08 04        1027              DB      4,100,8,4
AF51 04 64 08 04        1028              DB      4,100,8,4
AF55 04 64 08 04        1029              DB      4,100,8,4
AF59 04 64 08 04        1030              DB      4,100,8,4
AF5D 04 64 08 04        1031              DB      4,100,8,4
AF61 04 64 08 04        1032              DB      4,100,8,4
AF65 04 64 08 04        1033              DB      4,100,8,4
AF69 04 64 08 04        1034              DB      4,100,8,4
AF6D 04 64 08 04        1035              DB      4,100,8,4
AF71 04 64 08 04        1036              DB      4,100,8,4
AF75 04 64 08 04        1037              DB      4,100,8,4
AF79 04 64 08 04        1038              DB      4,100,8,4
AF7D 04 64 08 04        1039              DB      4,100,8,4
AF81 04 64 08 04        1040              DB      4,100,8,4
AF85 04 64 08 04        1041              DB      4,100,8,4
AF89 04 64 08 04        1042              DB      4,100,8,4
AF8D                    1043 ACTF
AF8D 00 00 00 00 00 00 00  1044           DS      16
AF94 00 00 00 00 00 00 00
AF9B 00 00
AF9D                    1045 PC
AF9D 00 00 00 00 00 00 00  1046           DS      16*2
AFA4 00 00 00 00 00 00 00
AFAB 00 00 00 00 00 00 00
AFB2 00 00 00 00 00 00 00
AFB9 00 00 00 00
AFBD                    1047 PANTBL
AFBD 0A                 1048              DB      10         ; Ctrl No.
AFBE 00                 1049              DB      00         ; Most LEFT
AFBF 0A                 1050              DB      10
AFC0 14                 1051              DB      20
AFC1 1E                 1052              DB      30         ; for D10
AFC2 28                 1053              DB      40         ; (Roland)
AFC3 32                 1054              DB      50
AFC4 3C                 1055              DB      60
AFC5 40                 1056              DB      64         ; CENTER
AFC6 50                 1057              DB      80
AFC7 55                 1058              DB      85
AFC8 5F                 1059              DB      95
AFC9 64                 1060              DB      100
AFCA 6E                 1061              DB      110
AFCB 78                 1062              DB      120
AFCC 7F                 1063              DB      127        ; Most RIGHT
AFCD                    1064 AFLAG
```

```
AFCD 00 00 00 00 00 00 00   1065              DS      16
AFD4 00 00 00 00 00 00 00
AFDB 00 00
AFDD                        1066 NTWORK
AFDD 00 00 00 00 00 00 00   1067              DS      16
AFE4 00 00 00 00 00 00 00
AFEB 00 00
AFED                        1068 MFLAG
AFED 00 00 00 00 00 00 00   1069              DS      16
AFF4 00 00 00 00 00 00 00
AFFB 00 00
AFFD                        1070 DESTACK
AFFD 00 00                  1071              DS      2
AFFF                        1072 PFLAG
AFFF 00                     1073              DS      1
B000                        1074              ;
B000                        1075 ;
B000                        1076 ;<<<<<<<<<<<<<< $3D6E ENTRY >>>>>>>>>>>>
B000                        1077 ;
3D6E                        1078              ORG     $3D6E
3D6E                        1079              ;
3D6E                        1080              OFFSET  $D000-$3D6E
3D6E                        1081              ;
3D6E                        1082 INST00
3D6E D5                     1083              PUSH    DE
3D6F CD FA AD               1084              CALL    NUMBER
3D72 DA DA 3D               1085              JP      C,INST9
3D75 7D                     1086              LD      A,L
3D76 B7                     1087              OR      A
3D77 28 0B                  1088              JR      Z,INST0
3D79 3D                     1089              DEC     A
3D7A FE E4                  1090              CP      228
3D7C D2 DA 3D               1091              JP      NC,INST9
3D7F FE 28                  1092              CP      40
3D81 D2 2A 3F               1093              JP      NC,NEIRO
3D84                        1094 INST0
3D84 F5                     1095              PUSH    AF
3D85 3E 01                  1096              LD      A,1
3D87 CD E3 3D               1097              CALL    WMFLAG
3D8A 7B                     1098              LD      A,E
3D8B FE 08                  1099              CP      8
3D8D 30 43                  1100              JR      NC,INST8
3D8F                        1101 INST01
3D8F F1                     1102              POP     AF
3D90                        1103              ;
3D90 D1                     1104              POP     DE
3D91 D5                     1105              PUSH    DE
3D92 21 39 40               1106              LD      HL,INSTN
3D95 19                     1107              ADD     HL,DE
3D96 77                     1108              LD      (HL),A
3D97                        1109              ;
3D97 6F                     1110              LD      L,A
3D98 CD 5B 3E               1111              CALL    GETVTD
3D9B                        1112              ;
3D9B D1                     1113              POP     DE
3D9C C5                     1114              PUSH    BC
3D9D E5                     1115              PUSH    HL
3D9E 3E 20                  1116              LD      A,$20
3DA0 83                     1117              ADD     A,E
3DA1 CD DE 3D               1118              CALL    WOPMX
3DA4 C6 10                  1119              ADD     A,$10
3DA6 16 14                  1120              LD      D,$14
3DA8 CD 3F AE               1121              CALL    WOPM
3DAB C6 08                  1122              ADD     A,8
3DAD CD DE 3D               1123              CALL    WOPMX
3DB0                        1124              ;
3DB0 1E 06                  1125              LD      E,6
3DB2                        1126 INST1
3DB2 CD 6A 3E               1127              CALL    WOPMXX
3DB5 1D                     1128              DEC     E
3DB6 20 FA                  1129              JR      NZ,INST1
3DB8                        1130              ;
3DB8 11 05 00               1131              LD      DE,5
3DBB 19                     1132              ADD     HL,DE
3DBC 3E 18                  1133              LD      A,$18
3DBE CD DE 3D               1134              CALL    WOPMX
3DC1 3C                     1135              INC     A
3DC2 CD DE 3D               1136              CALL    WOPMX
3DC5 CD DE 3D               1137              CALL    WOPMX
3DC8 3E 1B                  1138              LD      A,$1B
3DCA CD DE 3D               1139              CALL    WOPMX
3DCD E1                     1140              POP     HL
3DCE C1                     1141              POP     BC
3DCF C3 68 AB               1142              JP      ACT8
3DD2                        1143 INST8
3DD2 F1                     1144              POP     AF
3DD3 CD CC AE               1145              CALL    DFLTBL
3DD6 23                     1146              INC     HL
3DD7 3E 0C                  1147              LD      A,12     ; V = 12
3DD9 77                     1148              LD      (HL),A
3DDA                        1149 INST9
3DDA D1                     1150              POP     DE
3DDB C3 68 AB               1151              JP      ACT8
3DDE                        1152 WOPMX
3DDE 56                     1153              LD      D,(HL)
3DDF 23                     1154              INC     HL
3DE0 C3 3F AE               1155              JP      WOPM
3DE3                        1156 WMFLAG
3DE3 E5                     1157              PUSH    HL
3DE4 21 ED AF               1158              LD      HL,MFLAG
3DE7 19                     1159              ADD     HL,DE
3DE8 77                     1160              LD      (HL),A
3DE9 E1                     1161              POP     HL
3DEA C9                     1162              RET
3DEB                        1163              ;
3DEB                        1164 OPMV0
3DEB F5                     1165              PUSH    AF
3DEC D5                     1166              PUSH    DE
3DED CD CC AE               1167              CALL    DFLTBL
3DF0 23                     1168              INC     HL
3DF1 7E                     1169              LD      A,(HL)
3DF2 F5                     1170              PUSH    AF
3DF3 21 39 40               1171              LD      HL,INSTN
3DF6 19                     1172              ADD     HL,DE
3DF7 6E                     1173              LD      L,(HL)
3DF8                        1174              ;
3DF8 CD 5B 3E               1175              CALL    GETVTD
3DFB 7E                     1176              LD      A,(HL)
3DFC E6 07                  1177              AND     7
3DFE 0E 07                  1178              LD      C,7
3E00 FE 04                  1179              CP      4
3E02 38 0B                  1180              JR      C,OPMV1
3E04 CB 89                  1181              RES     1,C
3E06 28 07                  1182              JR      Z,OPMV1
3E08 CB 91                  1183              RES     2,C
3E0A FE 07                  1184              CP      7
3E0C 20 01                  1185              JR      NZ,OPMV1
3E0E 0D                     1186              DEC     C
3E0F                        1187 OPMV1
3E0F 11 06 00               1188              LD      DE,6
3E12 19                     1189              ADD     HL,DE
3E13 11 81 40               1190              LD      DE,V4
3E16 06 04                  1191              LD      B,4
3E18                        1192 OPMV2
3E18 7E                     1193              LD      A,(HL)
3E19 CB 09                  1194              RRC     C
3E1B 30 02                  1195              JR      NC,OPMV3
3E1D F6 80                  1196              OR      $80
3E1F                        1197 OPMV3
3E1F 12                     1198              LD      (DE),A
3E20 23                     1199              INC     HL
3E21 13                     1200              INC     DE
3E22 10 F4                  1201              DJNZ    OPMV2
3E24                        1202              ;
3E24 06 04                  1203              LD      B,4
3E26 EB                     1204              EX      DE,HL
3E27 2B                     1205              DEC     HL
3E28 3E 7F                  1206              LD      A,$7F
3E2A                        1207 OPMV4
3E2A BE                     1208              CP      (HL)
3E2B 38 01                  1209              JR      C,OPMV5
3E2D 7E                     1210              LD      A,(HL)
3E2E                        1211 OPMV5
3E2E 2B                     1212              DEC     HL
3E2F 10 F9                  1213              DJNZ    OPMV4
3E31                        1214              ;
3E31 C1                     1215              POP     BC
3E32 ED 44                  1216              NEG
3E34 90                     1217              SUB     B
3E35 C6 7F                  1218              ADD     A,127
3E37 4F                     1219              LD      C,A
3E38 D1                     1220              POP     DE
3E39 D5                     1221              PUSH    DE
3E3A 3E 58                  1222              LD      A,$58
3E3C 83                     1223              ADD     A,E
3E3D F5                     1224              PUSH    AF
3E3E 23                     1225              INC     HL
3E3F E5                     1226              PUSH    HL
3E40 06 04                  1227              LD      B,4
3E42                        1228 OPMV6
3E42 7E                     1229              LD      A,(HL)
3E43 B7                     1230              OR      A
3E44 FA 4D 3E               1231              JP      M,OPMV7
3E47 81                     1232              ADD     A,C
3E48 F2 4D 3E               1233              JP      P,OPMV7
3E4B 3E 7F                  1234              LD      A,$7F
3E4D                        1235 OPMV7
3E4D E6 7F                  1236              AND     $7F
3E4F 77                     1237              LD      (HL),A
3E50 23                     1238              INC     HL
3E51 10 EF                  1239              DJNZ    OPMV6
3E53                        1240              ;
3E53 E1                     1241              POP     HL
3E54 F1                     1242              POP     AF
3E55 CD 6A 3E               1243              CALL    WOPMXX
3E58                        1244              ;
3E58 D1                     1245              POP     DE
3E59 F1                     1246              POP     AF
3E5A C9                     1247              RET
3E5B                        1248 GETVTD
3E5B 26 00                  1249              LD      H,0
3E5D 29                     1250              ADD     HL,HL
3E5E 29                     1251              ADD     HL,HL
3E5F 54 5D                  1252              LD      DE,HL
3E61 29                     1253              ADD     HL,HL
3E62 29                     1254              ADD     HL,HL
3E63 29                     1255              ADD     HL,HL
3E64 19                     1256              ADD     HL,DE
3E65 11 90 B1               1257              LD      DE,$B190
3E68 19                     1258              ADD     HL,DE
3E69 C9                     1259              RET
3E6A                        1260 WOPMXX
3E6A C6 08                  1261              ADD     A,$08
3E6C CD DE 3D               1262              CALL    WOPMX
3E6F C6 10                  1263              ADD     A,$10
3E71 CD DE 3D               1264              CALL    WOPMX
3E74 D6 08                  1265              SUB     $08
3E76 CD DE 3D               1266              CALL    WOPMX
3E79 C6 10                  1267              ADD     A,$10
3E7B CD DE 3D               1268              CALL    WOPMX
3E7E C9                     1269              RET
3E7F                        1270              ;
3E7F                        1271 INISIO
3E7F F5                     1272              PUSH    AF
3E80 C5                     1273              PUSH    BC
3E81 E5                     1274              PUSH    HL
3E82                        1275              ;
3E82 3E 09                  1276              LD      A,9
3E84 01 01 00               1277              LD      BC,$0001
3E87 21 94 3E               1278              LD      HL,ISTBL
3E8A                        1279 INISIO2
3E8A 04                     1280              INC     B
3E8B ED A3                  1281              OUTI
3E8D 3D                     1282              DEC     A
3E8E 20 FA                  1283              JR      NZ,INISIO2
3E90 E1                     1284              POP     HL
3E91 C1                     1285              POP     BC
3E92 F1                     1286              POP     AF
3E93 C9                     1287              RET
3E94                        1288 ISTBL
3E94 18 01 00 03 C1         1289              DB      $18:$01:$00:$03:$C1
3E99 04 44 05 68            1290              DB      $04:$44:$05:$68
3E9D                        1291              ;
3E9D                        1292 MIDI0
3E9D E6 7F                  1293              AND     $7F
3E9F                        1294 MIDI
3E9F C5                     1295              PUSH    BC
3EA0 F5                     1296              PUSH    AF
3EA1 01 01 00               1297              LD      BC,$0001
3EA4                        1298 MIDI2
3EA4 ED 78                  1299              IN      A,(C)
3EA6 CB 57                  1300              BIT     2,A
3EA8 28 FA                  1301              JR      Z,MIDI2
3EAA F1                     1302              POP     AF
3EAB 0B                     1303              DEC     BC
3EAC ED 79                  1304              OUT     (C),A
3EAE C1                     1305              POP     BC
3EAF C9                     1306              RET
3EB0                        1307 NOTEON
3EB0 F5                     1308              PUSH    AF
3EB1 CD 1A 40               1309              CALL    ANDCHECK
3EB4 20 11                  1310              JR      NZ,NOTEON2 ; NZ=&
3EB6 F1                     1311              POP     AF
3EB7 CD E3 3E               1312              CALL    NTNOTE
3EBA 77                     1313              LD      (HL),A
3EBB CD D5 3E               1314              CALL    NOTEOUT
3EBE CD CC AE               1315              CALL    DFLTBL
3EC1 23                     1316              INC     HL
3EC2 7E                     1317              LD      A,(HL)     ; A=Vol
3EC3 CD 9F 3E               1318              CALL    MIDI
3EC6 C9                     1319              RET
3EC7                        1320 NOTEON2
3EC7 F1                     1321              POP     AF
3EC8 C9                     1322              RET
3EC9                        1323 NOTEOFF
3EC9 CD E3 3E               1324              CALL    NTNOTE
3ECC 7E                     1325              LD      A,(HL)
3ECD CD D5 3E               1326              CALL    NOTEOUT
3ED0 AF                     1327              XOR     A          ; Vol=0
3ED1 CD 9F 3E               1328              CALL    MIDI
3ED4 C9                     1329              RET
3ED5                        1330 NOTEOUT
3ED5 F5                     1331              PUSH    AF         ; IN = A
```

```
3ED6 CD 22 3F         1332          CALL    MIDICH
3ED9 C6 90            1333          ADD     A,$90
3EDB CD 9F 3E         1334          CALL    MIDI
3EDE F1               1335          POP     AF
3EDF CD 9F 3E         1336          CALL    MIDI
3EE2 C9               1337          RET
3EE3                  1338 NTNOTE
3EE3 16 00            1339          LD      D,0
3EE5 21 DD AF         1340          LD      HL,NTWORK
3EE8 19               1341          ADD     HL,DE
3EE9 C9               1342          RET
3EEA                  1343          ;
3EEA                  1344 NTNUM
3EEA 7D               1345          LD      A,L
3EEB D5               1346          PUSH    DE      ;IN = L (A-G 0-14)
3EEC F5               1347          PUSH    AF      ;     E (CH. NO.)
3EED CD CC AE         1348          CALL    DFLTBL ;OUT= A (NOTE NO.)
3EF0 7E               1349          LD      A,(HL)
3EF1 3C               1350          INC     A
3EF2 07               1351          RLCA
3EF3 07               1352          RLCA
3EF4 5F               1353          LD      E,A
3EF5 07               1354          RLCA
3EF6 83               1355          ADD     A,E         ;A*12
3EF7 5F               1356          LD      E,A
3EF8                  1357          ;
3EF8 F1               1358          POP     AF          ;A = 9
3EF9 FE 05            1359          CP      5           ;B =11
3EFB 30 02            1360          JR      NC,NTNUM2 ;C = 0
3EFD C6 0E            1361          ADD     A,14        ;D = 2
3EFF                  1362 NTNUM2                 ;E = 4
3EFF FE 0B            1363          CP      11          ;F = 5
3F01 38 01            1364          JR      C,NTNUM3    ;G = 7
3F03 3D               1365          DEC     A
3F04                  1366 NTNUM3
3F04 D6 05            1367          SUB     5
3F06                  1368          ;
3F06 83               1369          ADD     A,E
3F07 D1               1370          POP     DE
3F08 C9               1371          RET
3F09                  1372          ;
3F09                  1373 MIDNUM
3F09 CD FA AD         1374          CALL    NUMBER      ; " Ma "
3F0C 38 11            1375          JR      C,MIDNUM2 ; IN = A(Ch.No)
3F0E 7D               1376          LD      A,L         ;        1-16
3F0F FE 11            1377          CP      17
3F11 30 0C            1378          JR      NC,MIDNUM2
3F13 B7               1379          OR      A
3F14 28 09            1380          JR      Z,MIDNUM2
3F16 3D               1381          DEC     A
3F17 16 00            1382          LD      D,0
3F19 21 29 40         1383          LD      HL,CHTBL
3F1C 19               1384          ADD     HL,DE
3F1D 77               1385          LD      (HL),A
3F1E 3C               1386          INC     A          ; Z = 0
3F1F                  1387 MIDNUM2
3F1F C3 68 AB         1388          JP      ACT8
3F22                  1389 MIDICH
3F22 E5               1390          PUSH    HL          ;GET MIDI CH.
3F23 21 29 40         1391          LD      HL,CHTBL    ;OUT = A(Ch.No)
3F26 19               1392          ADD     HL,DE
3F27 7E               1393          LD      A,(HL)
3F28 E1               1394          POP     HL
3F29 C9               1395          RET
3F2A                  1396          ;
3F2A                  1397 NEIRO
3F2A D1               1398          POP     DE
3F2B FE 64            1399          CP      100
3F2D DA 68 AB         1400          JP      C,ACT8
3F30 D6 64            1401          SUB     100         ; " Ia "
3F32 6F               1402          LD      L,A         ; OUT $Cn,a
3F33 CD 22 3F         1403          CALL    MIDICH
3F36 C6 C0            1404          ADD     A,$C0
3F38 CD 9F 3E         1405          CALL    MIDI
3F3B 7D               1406          LD      A,L
3F3C CD 9D 3E         1407          CALL    MIDI0
3F3F AF               1408          XOR     A
3F40 CD E3 3D         1409          CALL    WMFLAG
3F43 3C               1410          INC     A          ; Z = 0
3F44 C3 68 AB         1411          JP      ACT8
3F47                  1412 TARE
3F47 ED 78            1413          IN      A,(C)  ; " Z(a,b,c,) "
3F49 FE 28            1414          CP      '('    ; OUT =a,b,c
3F4B 28 09            1415          JR      Z,TARE2; " Z[a,b,c] "
3F4D FE 5B            1416          CP      '['    ; OUT =F0,a,b,c,F7
3F4F 20 1E            1417          JR      NZ,TARE4
3F51 3E F0            1418          LD      A,$F0
3F53 CD 9F 3E         1419          CALL    MIDI
3F56                  1420 TARE2
3F56 03               1421          INC     BC
3F57 CD FA AD         1422          CALL    NUMBER
3F5A 38 0A            1423          JR      C,TARE3
3F5C 7D               1424          LD      A,L
3F5D CD 9F 3E         1425          CALL    MIDI
3F60 ED 78            1426          IN      A,(C)
3F62 FE 2C            1427          CP      ','
3F64 28 F0            1428          JR      Z,TARE2
3F66                  1429 TARE3
3F66 FE 5D            1430          CP      ']'
3F68 20 05            1431          JR      NZ,TARE4
3F6A 3E F7            1432          LD      A,$F7
3F6C CD 9F 3E         1433          CALL    MIDI
3F6F                  1434 TARE4
3F6F 03               1435          INC     BC
3F70 C3 68 AB         1436          JP      ACT8
3F73                  1437 CTRLC
3F73 CD FA AD         1438          CALL    NUMBER      ; " ^a,b "
3F76 38 21            1439          JR      C,CTRLC2    ; OUT $Bn,a,b
3F78 ED 78            1440          IN      A,(C)
3F7A FE 2C            1441          CP      ','
3F7C 20 1B            1442          JR      NZ,CTRLC2
3F7E E5               1443          PUSH    HL
3F7F 03               1444          INC     BC
3F80 CD FA AD         1445          CALL    NUMBER
3F83 7D               1446          LD      A,L
3F84 E1               1447          POP     HL
3F85 38 12            1448          JR      C,CTRLC2
3F87 67               1449          LD      H,A
3F88 CD 22 3F         1450          CALL    MIDICH
3F8B C6 B0            1451          ADD     A,$B0
3F8D CD 9F 3E         1452          CALL    MIDI
3F90 7D               1453          LD      A,L
3F91 CD 9D 3E         1454          CALL    MIDI0
3F94 7C               1455          LD      A,H
3F95 CD 9D 3E         1456          CALL    MIDI0
3F98 3C               1457          INC     A          ; Z = 0
3F99                  1458 CTRLC2
3F99 C3 68 AB         1459          JP      ACT8
3F9C                  1460 BEND
3F9C CD FA AD         1461          CALL    NUMBER      ; " ¥a "
3F9F 38 19            1462          JR      C,BEND2     ; OUT $En,?,?
3FA1 AF               1463          XOR     A
3FA2 CB 1D            1464          RR      L
3FA4 CB 1F            1465          RR      A
3FA6 CB 1F            1466          RR      A
3FA8 F5               1467          PUSH    AF
3FA9 CD 22 3F         1468          CALL    MIDICH
3FAC C6 E0            1469          ADD     A,$E0
3FAE CD 9F 3E         1470          CALL    MIDI
3FB1 F1               1471          POP     AF
3FB2 CD 9D 3E         1472          CALL    MIDI0
3FB5 7D               1473          LD      A,L
3FB6 CD 9D 3E         1474          CALL    MIDI0
3FB9 3C               1475          INC     A          ; Z = 0
3FBA                  1476 BEND2
3FBA C3 68 AB         1477          JP      ACT8
3FBD                  1478 STCTC
3FBD CD FA AD         1479          CALL    NUMBER
3FC0 38 17            1480          JR      C,STCTC2
3FC2 7D               1481          LD      A,L
3FC3 32 D8 AE         1482          LD      (TMPV),A
3FC6 ED 78            1483          IN      A,(C)
3FC8 FE 2C            1484          CP      ','
3FCA 20 0D            1485          JR      NZ,STCTC2
3FCC 03               1486          INC     BC
3FCD CD FA AD         1487          CALL    NUMBER
3FD0 38 07            1488          JR      C,STCTC2
3FD2 7D               1489          LD      A,L
3FD3 32 D7 AE         1490          LD      (TMPV0),A
3FD6 CD 2E AA         1491          CALL    TOCTC
3FD9                  1492 STCTC2
3FD9 C3 68 AB         1493          JP      ACT8
3FDC                  1494 MIDIPAN
3FDC CD FA AD         1495          CALL    NUMBER
3FDF 7D               1496          LD      A,L
3FE0 B7               1497          OR      A
3FE1 28 1B            1498          JR      Z,MPAN2
3FE3 FE 10            1499          CP      16
3FE5 30 18            1500          JR      NC,MPAN3
3FE7 F5               1501          PUSH    AF
3FE8 CD 22 3F         1502          CALL    MIDICH
3FEB C6 B0            1503          ADD     A,$B0
3FED CD 9F 3E         1504          CALL    MIDI
3FF0 21 BD AF         1505          LD      HL,PANTBL
3FF3 7E               1506          LD      A,(HL)
3FF4 CD 9D 3E         1507          CALL    MIDI0
3FF7 F1               1508          POP     AF
3FF8 85               1509          ADD     A,L
3FF9 6F               1510          LD      L,A
3FFA 7E               1511          LD      A,(HL)
3FFB CD 9D 3E         1512          CALL    MIDI0
3FFE                  1513 MPAN2
3FFE 3C               1514          INC     A
3FFF                  1515 MPAN3
3FFF C3 68 AB         1516          JP      ACT8
4002                  1517          ;
4002                  1518 MDCHECK
4002 E5               1519          PUSH    HL
4003 21 ED AF         1520          LD      HL,MFLAG
4006 19               1521          ADD     HL,DE
4007 7E               1522          LD      A,(HL)
4008 E1               1523          POP     HL
4009 B7               1524          OR      A
400A C9               1525          RET
400B                  1526 ANDFLAG0
400B AF               1527          XOR     A
400C                  1528 ANDFLAG
400C D5               1529          PUSH    DE
400D E5               1530          PUSH    HL
400E ED 5B FD AF      1531          LD      DE,(DESTACK)
4012 21 CD AF         1532          LD      HL,AFLAG
4015 19               1533          ADD     HL,DE
4016 77               1534          LD      (HL),A
4017 E1               1535          POP     HL
4018 D1               1536          POP     DE
4019 C9               1537          RET
401A                  1538 ANDCHECK
401A D5               1539          PUSH    DE
401B E5               1540          PUSH    HL
401C ED 5B FD AF      1541          LD      DE,(DESTACK)
4020 21 CD AF         1542          LD      HL,AFLAG
4023 19               1543          ADD     HL,DE
4024 7E               1544          LD      A,(HL)
4025 E1               1545          POP     HL
4026 D1               1546          POP     DE
4027 B7               1547          OR      A
4028 C9               1548          RET
4029                  1549 ;
4029                  1550 ;<<<<<<<<<<<<<<< WORK AREA 2 >>>>>>>>>>>>
4029                  1551 ;
4029                  1552 CHTBL
4029 00 01 02 03      1553          DB      $00:$01:$02:$03
402D 04 05 06 07      1554          DB      $04:$05:$06:$07
4031 08 09 0A 0B      1555          DB      $08:$09:$0A:$0B
4035 0C 0D 0E 0F      1556          DB      $0C:$0D:$0E:$0F
4039                  1557 INSTN
4039 01 01 01 01      1558          DB      1,1,1,1 ;for OPM
403D 01 01 01 01      1559          DB      1,1,1,1
4041                  1560 PACK
4041 00 00 00 00 00 00 00   1561    DS      16*4
4048 00 00 00 00 00 00 00
404F 00 00 00 00 00 00 00
4056 00 00 00 00 00 00 00
405D 00 00 00 00 00 00 00
4064 00 00 00 00 00 00 00
406B 00 00 00 00 00 00 00
4072 00 00 00 00 00 00 00
4079 00 00 00 00 00 00 00
4080 00
4081                  1562 V4
4081 00 00 00 00      1563          DS      4
4085                  1564 ;
4085                  1565 ;<<<<<<<<<<<<<<< TIMING CLOCK >>>>>>>>>>>
4085                  1566 ;
4085                  1567 TCLOCK
4085 E5               1568          PUSH    HL
4086 21 98 40         1569          LD      HL,TCWORK
4089 35               1570          DEC     (HL)
408A 20 0A            1571          JR      NZ,TCLOCK2
408C F5               1572          PUSH    AF
408D 3E F8            1573          LD      A,$F8    ; T-Clock
408F CD 9F 3E         1574          CALL    MIDI
4092 3E 04            1575          LD      A,4
4094 77               1576          LD      (HL),A
4095 F1               1577          POP     AF
4096                  1578 TCLOCK2
4096 E1               1579          POP     HL
4097 C9               1580          RET
4098                  1581 TCWORK
4098 04               1582          DB      4
4099                  1583 END40E8
4099 00 00 00 00 00 00 00   1584    DS      $40E8-END40E8
40A0 00 00 00 00 00 00 00
40A7 00 00 00 00 00 00 00
40AE 00 00 00 00 00 00 00
40B5 00 00 00 00 00 00 00
40BC 00 00 00 00 00 00 00
40C3 00 00 00 00 00 00 00
40CA 00 00 00 00 00 00 00
40D1 00 00 00 00 00 00 00
40D8 00 00 00 00 00 00 00
40DF 00 00 00 00 00 00 00
40E6 00 00
```

Z80マシン語
ゲーム工房

第1回

目指せシューティング

Murata Toshiyuki 村田 敏幸

今月からスタートするこの「Z80マシン語ゲーム工房」は，ゲームプログラムをサンプルとして取り上げながらマシン語の基礎を学んでいこうというものです。そしてそのゲームプログラムに関しては，皆さんからの要望もバシバシ取り入れていく予定らしいので，ぜひ参加してくださいね。

ご挨拶だよ

初めまして。今月からしばらくの間，Z80のマシン語で遊ぼうというみんなのお相手を務めるのがこの僕，村田だ。

Z80の入門講座といえば，今年の5月号で惜しまれつつ終了した泉大介氏の「マシン語体操1·2·3」を読んでいた人は多いだろう。「マシン語体操」は，全機種共通システムS-OS“SWORD”の上で厳選された命令のみを使用したプログラム例を示し，それに泉氏の見事な解説を加えることで，誰にでも読め，楽しめ，わかるという近年まれに見る名講座だった。遅ればせながら泉さんには「ごくろうさまでした」と言いたい。

と，いきなり殊勝な書き出しをしたのにはわけがある。というのは，僕には「マシン語体操」を超えるZ80入門講座は書けないかもしれない。ただ，違う方向性を示すことはできるだろう，との思いがあってこの連載を始めることにした。

マシン語を使う楽しみのひとつは，自分のマシンに直接触れることにある。これまでの「マシン語体操」の場合，S-OS上で扱っている分だけ「マシンにじかに触れる」感じが薄れてしまっていたように思う。そこでこの連載では，各機種別の基本レベルでの処理まで言及してみようと考えている。

誤解のないように言っておくが，機種別の要素を盛り込むからといって，Z80という共通の土台を無視してかかるわけではもちろんない。X1もMZもZ80マシンなのだから，プログラムの大筋は共通のものとなる。どうしてもハードに依存せざるを得ないとき，またはハードに依存したほうがよいときには別々に用意するよ，と言っているに過ぎない。その場合でも，やりたいことが同じである限り，アルゴリズムはほぼ共通のものになるだろう。「マシン語体操」とは異なる方向性と言いながらも，「なぁんだ，結局みんな同じZ80マシンなんだね」というところに帰って来るような気がしている。多少遠回りにはなるかもしれないけれど……。

で，具体的になにをやるかというと

横スクロールシューティングゲーム

を作ることにした。対象機種はX1，MZ-2500，MZ-700とする。当然X1版はturboで，MZ-700版は1500でも動作するから，Oh!Xを読んでいるほとんどの機種のユーザーには楽しんでもらえると思う。ここで名前を挙げなかったマシン(特にMZ-2000)も気が向けばサポートするつもりでいるし，turboと1500に関しては，それぞれの特徴を生かした変更点を示す用意はある。待っていればそのうちいいこともあるだろう。どうしても待てない人は「僕のマシンもやって係」まで怒濤のハガキ攻撃をかけてほしい。

これだけは用意してね

この連載ではダンプリストを掲載しない方針だ。それはダンプリストを打ち込むのがうまくなったからといって，マシン語を覚えられるわけではないという，当たり前の理由と（ゲームは少しずつ組み立てていく予定なので），ダンプリストでは毎月つじつまを合わせるのが困難だという現実的な理由による。また，限られたページをダンプリストで埋めてしまうのはもったいないとも考えた。こういったわけで，この連載を活用しようと思ったら，アセンブラを用意してもらう必要がある。

アセンブラはZ80用（ザイログ表記）のものであればなんでも構わないのだが，掲載されるアセンブルリストはOh!X標準アセンブラであるZEDA用のものであり，また，短いサンプルはS-OS“SWORD”上で動作するように作られるので，そのほかのシステムを使う人は注意を要する。S-OS“SWORD”とZEDAを用意してもらえればいちばんよい。

もうひとつ，Z80の命令表はどの道いつかは必要になるだろうから，いまのうちに用意しておくことが望ましい。アセンブラと命令表，これだけあれば当分の用には足りる（僕の場合は，未だにこれだけで足りていたりする)。

というところで，ぼちぼち本編に突入する。「わーい，いよいよゲームだぞぉ」と喜んだ人には申し訳ないが，今月と来月はゲームを作るために必要なZ80の基礎の部分をやる。やるといったらやる。

ダダをこねる子には「基礎を知らずしてゲームを作るとは2カ月早いっ！」という言葉を贈りたい。

マシン語は難しいの？

このテのマシン語入門では，最初に「マシン語は難しくありません」といって読者を安心させるのが常となっている。けれど，僕はへそ曲がりだからそうは言わない。どう言うかというと，マシン語を覚えるのは

BASICを覚えるのと同じぐらい難しい

と言う。別な言い方をすると，

逆上がりができるようになるくらい
飛び箱が飛べるようになるくらい
自転車に乗れるようになるくらい
泳げるようになるくらい

難しいとも言える。

要はコツの問題，きっかけの問題だ。また，よく「マシン語は暴走するから怖い」と言うけど，僕には暴走するとどうして怖いのかがよくわからない。そりゃあ，暴走したせいで作りかけのプログラムが消えてしまうこともあるし，ディスクを飛ばしてしまうこともある。でも，消えてしまったプログラムもセーブしてあればそこまでは戻ることはできるし，ディスクもバックアップをとってあれば困らない。そもそも暴走が原因でコンピュータが壊れてしまったりはしないんだから恐れることはないと思う。むしろ「暴走はしないが，なんとなく変な動作をする」バグが入り込むことのほうがよっぽど怖い。

暴走はして当然なんだ。僕だって毎日のように暴走させている。

それでは本題です

では，そろそろマシン語プログラムの実例を見てもらい，そこで使われている命令について解説していくことにする。マシン語の各命令は驚くほど単純な機能しか持っていないから，理解するのは造作もないことのはずだ。なお，本文中では平気な顔をして専門用語を使っていたり，詳しい説明をはしょっている箇所がある。それらについてはページのそこいらじゅうに解説をちりばめてあるので，用語の説明はそちらを参照してほしい。

まずはリスト1を見てもらおう。こいつはS-OS“SWORD”上で動作するごく簡単なプログラムで，アセンブルし実行すると画面にひとつ“Z”の文字を書く。ソースでは7行だが，1～4行目はZ80の命令ではなく，アセンブラに対する指令（疑似命令）だから，プログラム本体はわずか3行，3命令ということになる。

まずはこの3つの命令の働きから説明しよう。

5行目の

LD　　A,90

は，BASICでいう代入文にあたる。LDはLoaDの略だ。いまの例では「Aに90という値を入れる」という意味になる。Aが何者かはあとで説明することにする。ここではBASICでいう変数みたいなものだと思っていてくれれば十分だ。

続いて6行目の

CALL　　#PRINT

は，CALLの名のとおりサブルーチンを「呼び出して」いる。BASICであればGOSUBにあたる命令だ。CALLの後ろにはサブルーチンの始まるアドレスを書く。リスト1ではアドレスを数字でそのまま書かずにラベルを使っている。“#PRINT”というラベルには3行目にあるEQU疑似命令で，$1FF4_H$という値が与えられているから，6行目は

CALL　　1FF4H

と書いたのと同じ意味，つまり$1FF4_H$番地から始まるサブルーチンを呼び出していることになる。

なお，$1FF4_H$番地からはS-OS“SWORD”の1文字表示サブルーチンが置かれていて，そのサブルーチンは「A」に表示したい文字のアスキーコードを入れて呼び出すように作られている。5行目でAに入れた値90は“Z”のアスキーコードに相違ない。

最後の

RET

はRETurnの略で，サブルーチンから呼び出し元へ戻る命令だ。が，ここではサブルーチンから戻るのではなく，「プログラムの実行を終えてS-OSのモニタへ戻る」という働きをしている。ちょっと引っ掛かるものを感じるかもしれないが，このプログラム全体もS-OSから見ればサブルーチンのようなものとして扱われている，と考えておいてほしい。

さて，わずか数十行でZ80の3つの命令の働きを説明した。まだ「Aとはなにか」といった謎が残ってはいるが，LD, CALL, RETの3命令の働きはわかってもらえたことと思う。マシン語はこのくらいシンプルなんだ。

次の節ではさらにAの正体が解き明かされるだろう。

レジスタとメモリ

Aは変数のようなものだ，と言った。この「変数のようなもの」のことをマシン語ではレジスタという。図1にZ80の全レジスタを示す。いまはこのうちA, B, C, D, E, H, Lの7つだけを覚えておけばよい。

A～Lのレジスタはちょうどメモリの1番地分と同じ大きさ，つまり1バイトの大きさを持つ。1バイト＝8ビットだから，各レジスタには2進8桁（＝16進2桁）までの整数を格納することができることになる。具体的には

00000000_B～11111111_B

16進数では

00_H～FF_H

10進数に直すと

0～255

までの値が入る。

また，BASICではRUNしたときに変数

リスト1

```
8000            1         ORG     8000H
8000            2 ;
8000            3 #PRINT  EQU     1FF4H
8000            4 ;
8000 3E 5A      5         LD      A,90
8002 CD F4 1F   6         CALL    #PRINT
8005 C9         7         RET
```

アセンブラ

Oh!Xの読者には当てはまらないと思うが，一般にマシン語には「わけのわからない16進数の羅列」というイメージがある。で，その「わけのわからない～」を覚えなければマシン語でプログラムが作れないと思ってしまうらしい。マシン語でプログラムを作る＝16進数を並べることだと思っている人もいるそうな（さすがにここまで言ってしまうとウソくさいか）。

確かに最終的に，CPUが実行可能なのは「16進数の羅列」だし，ワンボードマイコンの時代には直接16進数を並べてプログラムを作ったらしい。が，この文明開化の時代にそんなマヌケな話があるわけはないじゃないか。実際にはマシン語プログラムは「アセンブリ言語」で記述・開発される。

アセンブリ言語はマシン語の命令とほぼ1対1で対応したシンプルな記号言語だ。アセンブリ言語で記述されたプログラムテキスト自体は，普通のアスキーファイルだから，むろん直接実行することはできない。実行できるようにするには例の「16進数」に変換する「アセンブル」という手順を踏む必要がある。そのアセンブルをしてくれるプログラムが「アセンブラ」だ。

また，アセンブリ言語で書かれたテキストは，マシン語プログラムの「基」となるものという意味で「アセンブリソース」といい，アセンブルしてできた実行可能なプログラムを「オブジェクト」と呼ぶ。

アセンブラには，すぐに実行できる形式のオブジェクト（早い話が，普通にいうマシン語プログラムのこと）を多くの場合メモリ上に直接生成する「アブソリュートアセンブラ」と，一度中間形式のオブジェクトをファイルに出力し，「リンカ」というプログラムを通して初めて実行可能となる「リロケータブルアセンブラ」がある。

一般的にアブソリュートアセンブラのほうがアセンブル速度が速いが，オブジェクトをメモリ上に生成する場合，大きなプログラムが作れないという欠点がある。逆にリロケータブルアセンブラはオブジェクトをファイルとして生成し，リンカを通さなければならないので実行プログラムを作るまでの時間はかかるが，プログラムを部品ごとに開発し，あとからいくつものオブジェクトをつなげてひとつのプログラムにすることができるので，大規模開発に向いている。

なお，かつてはアセンブラを使わずに命令表を片手に手作業でアセンブルする（ハンドアセンブルという）ことが，マシン語を覚える早道のような言われ方をしたことがあった。しかし，これは「足腰を鍛えるにはうさぎ跳びがいちばんだ」というのと同じくらい間違っている，と僕は思う。

ハンドアセンブルではプログラミングの本質とはまったく掛け離れたところに気を使わなければならないし，人間のやることだから勘違いからくる不必要なバグが入り込む恐れがある。その点アセンブラを使えばプログラミングそのものに集中できるわけだ。

マシン語はアセンブリ言語で記述し，アセンブラを使うことで初めて「プログラミング言語」となる。マシン語を覚えるにはアセンブラは必要不可欠だ，というのが僕の見解だ。

の内容がクリアされるが，レジスタはそうではないので，プログラムでちゃんと値を入れるまでは「いくつになっているか」がわからない。0になっていると決め込んで使ったりするとひどい目に遭う。注意すること。

A～Lまでのレジスタは「ほぼ」同じように使うことができる。リスト1ではAレジスタに直接数値を代入する例を示したが，同様にBレジスタに値を入れるには

```
LD      B,123
```

のように書けばよい。

また，LD命令を使ってレジスタの値をほかのレジスタに代入することもできる。Hレジスタに入っている値をAレジスタにコピーするには

```
LD      A,H
```

とする。簡単，簡単。

ここまでの例では値は常にレジスタに格納してきたが，メモリに値を格納することだってできる。メモリはレジスタのように名前がついていないから，任意のメモリはアドレスで指定することになる。

当然，メモリの内容をレジスタに持ってくることもできて，

```
LD      A,(9000H)
```

のように書けば，9000_H番地の内容がAレジスタに代入される。カッコが付いているのは，ただ

```
LD      A,9000H
```

と書いてしまうと，9000_H番地の内容ではなく，9000_Hという数をAに入れることになってしまうためだ（しかも，Aには16進2桁までしか入らないから，実際には9000_Hを入れることはできない）。このようにZ80のアセンブリ言語では「ある番地に格納されている値」を表すときにはカッコを付けることになっている。

逆にAレジスタの内容をメモリにしまっておきたければ，

```
LD      (9000H),A
```

のようにする。

アドレッシングモード

前述したように，同じLD命令でも後ろのほうをちょこちょこと変えることで操作の対象が違ってくる。この「操作対象の指定」のことを「アドレッシングモード」という。

```
LD      A,90
```

のように「ただの値（即値）」を指定する形式を「イミディエイトアドレッシング」といい，

```
LD      A,H
```

のようにレジスタを直接指定する形式のことを「レジスタ直接アドレッシング」，さらに，

```
LD      A,(9000H)
```

のようにメモリ番地を直接指定する形式を「絶対アドレッシング」と呼ぶ。

レジスタ直接があるのだから「レジスタ間接」もあるんじゃないかとか，絶対アドレッシングがあるからには「相対アドレッシング」もあるに違いないと考えた人はいい勘をしている。これらについてはいずれ実例が出てきたところで説明しよう。

ところでBASICの場合，たとえばPRINT文の後ろにくるのが変数だろうが，ただの数字だろうが，配列だろうが構わなかったわけで，アドレッシングモードに相当する概念は存在しなかった。その意味では，ずっとBASICを使ってきた人にはアドレッシングモードなんてのは非常に奇異なものに見えるだろう。LD命令は値を代入する命令だ，と知っていれば，アドレッシングモードなんて覚えなくても

```
LD      なんとか，かんとか
```

と書けばいいように思うかもしれない。

が，残念なことにマシン語（特にZ80）ではこの考え方は通用しない。そのココロは，「マシン語ではアドレッシングモードが違うだけで，CPUにとっては別の命令とみなされる」という点にある。もう少し言うと，「CPUにしてみれば違う命令なのだが，同じ働きの命令は同じ記号で表したほうが，プログラムする人間にとってわかりやすいからアセンブリ言語では同じ記号を使うことにした」というだけのことだ。本当は，

```
でべでべ  9000H
```

と書けば

```
LD      A,(9000H)
```

の意味で，

```
あわあわ  9000H
```

と書けば

```
LD      (9000H),A
```

の意味になる，というように決めてもよかったけれども（事実Z80の前身である8080というCPU用のアセンブリ言語では似たようなことをしている），使い分けるのが面倒だからLDに統一されていると考えてほしい。

そして，ここが肝心なところなのだが，「ほんとは違う命令に同じ記号を割り当てた」ために，「一見ありそうな命令が存在しない」ということが起こる。たとえば

```
LD      (9000H),(9001H)
```

のような形式はないし，

```
LD      (9000H),45
```

のような形もない。さらには，アドレッシングモードとレジスタの組み合わせにも制限があり，

```
LD      (9000H),A
LD      A,(9000H)
```

はあっても

```
LD      (9000H),B
LD      B,(9000H)
```

はない。どの組み合わせがあって，どれがないのかは命令表を参照してほしい。ただ，Aレジスタはかなり特別扱いされていて，命令にもよるが，ほとんどのアドレッシングモードとの組み合わせがサポートされている。

計算してみよう

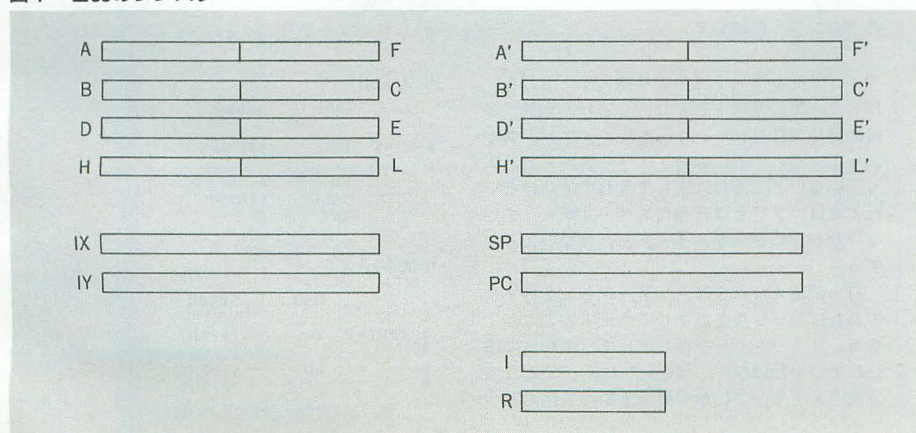
前節でAレジスタは特別なレジスタだと言った。どのくらい特別かというと，わざわざ「アキュムレータ(加算機)」という名前がついているぐらい特別だ。その名が示すように，Aレジスタは計算をするときには重要な地位にある。

Z80にできる演算は足し算，引き算，A

図1　Z80のレジスタ

A [　　|　　] F　　A' [　　|　　] F'
B [　　|　　] C　　B' [　　|　　] C'
D [　　|　　] E　　D' [　　|　　] E'
H [　　|　　] L　　H' [　　|　　] L'

IX [　　　　]　　SP [　　　　]
IY [　　　　]　　PC [　　　　]

I [　　]
R [　　]

ND(論理積)，OR(論理和)，XOR(排他的論理和) しかない。これらはアセンブリ言語ではそれぞれ

ADD
SUB
AND
OR
XOR

という命令で表され，(8ビット演算では)「Aレジスタになんかを足して結果をAレジスタに入れる」，「Aレジスタからなんかを引いて結果をAレジスタに入れる」というように，必ずAレジスタを使って計算することになっている。

リスト2を見てもらおう。このプログラムはAレジスタとBレジスタに適当な数を入れておいて，足した結果（Aレジスタ）を16進数で表示するものだ。例によって，「Aレジスタを16進数で表示する」処理はS-OS“SWORD”のサブルーチンを利用している。また，リスト1では“Z”を表示したあと改行していなかったが，今度は最後（14行）にS-OSの改行するサブルーチン（#LTNL）を呼び出すようにしてある。

しばらくはこのプログラムでレジスタに入れる値を変えたり，8行目を9行以下に注釈で示したように変更して遊んでみてほしい。特に，論理演算がどのようなものか自信がない人は，レジスタに与えた値と結果をよく見比べてみるとよいだろう。遊び飽きたら（オーイ，ちゃんと試してみた？）次に進む。

リスト2を見て，または実際に試してみるうちに，いくつかの疑問が生まれたことと思う。まず，リストを見て気がつくのは

ADD　　A, B
SUB　　B
AND　　B
OR　　B
XOR　　B

というように，ADD命令だけには“A,”が付いているが，ほかの命令には付いていない点だろう。

すでに述べたように，8ビット演算では必ずAレジスタとの間で演算が行われる。だから，

SUB　　A, B

と書かずに

SUB　　B

と書くだけで「アセンブラにはちゃんと意味が伝わる」。逆の見方をすると，ADD命令は「Aレジスタではないもの」と足し算をすることもできるので，アセンブラに区別できるようにAを付けるんだ。それがなにかは来月説明する。

2番目の謎は，勘のよい人ならリスト2を実行する前に，勘が鈍くてもマメに試してみた人ならそのときに気づき，勘が鈍くてちゃんとリスト2で遊んでみなかった人はまだ気づいていないかもしれない。

A～Lのレジスタは8ビットの大きさで0～255の値を入れることができるんだった。では，Aが255のとき1を足すと，Aはいったいいくつになるんだろうか？　リスト2の6行目を

LD　　A, 255

7行目を

LD　　B, 1

として試してみよう。

結果は，

00

と表示されたはずだ。どうしてか？

255は16進数で書くと

FF_H

となり，1は16進数に直しても

1_H

のままだ。FF_Hに01_Hを足すと，本当は

$FF_H+01_H=100_H$

とならなければならない。ところが，Aレジスタには16進2桁の数までしか格納できないので，Z80は下2桁だけを残したというのが真相だ。はみ出した上の桁は「どこ

リスト2

```
8000              1         ORG     8000H
8000              2 ;
8000              3 #PRTHX  EQU     1FC1H
8000              4 #LTNL   EQU     1FEEH
8000              5 ;
8000 3E 12        6         LD      A,12H
8002 06 34        7         LD      B,34H
8004 80           8         ADD     A,B
8005              9 ;       SUB     B
8005             10 ;       AND     B
8005             11 ;       OR      B
8005             12 ;       XOR     B
8005 CD C1 1F    13         CALL    #PRTHX
8008 CD EE 1F    14         CALL    #LTNL
800B C9          15         RET
```

初めてのZEDA

ZEDAはS-OS上で動作するエディタアセンブラで，ソースファイルの作成からアセンブル，完成したオブジェクトの実行，セーブまでを行うことができる。アセンブラの項で述べた分類でいうとアブソリュートアセンブラで，メモリ上にあるソースをアセンブルし，メモリ上に直接オブジェクトを生成する。

いきなりだが，ごく簡単なソースプログラム，リストa)を打ち込んでもらおう。手順は次のようになる。

まず，S-OS“SWORD”のモニタ上から

#L ZEDA

としてZEDA をロードし，

#J3000

でZEDAを起動する。起動時はアセンブラモードだから

E

でエディタモードに入り，さらに

E>

のプロンプトに続いて

E>I

でインサートモード（つまり挿入モードだな）にし，一気にリストどおり打ち込む。当然，各行の最後ではリターンキーを押す。スペースが入っているところには必ずスペースを入れ，入っていないところには決して入れてはいけない。たとえば，3行目は行頭にスペースが入っていないから頭に余分なスペースを入れないようにする。

打ち間違えたときは，リターンキーを押す前であればカーソルを動かして修正することができる。リターンキーを押してしまってから間違いに気づいたのなら，「放っておく」こと。多少間違えても，とにかく最後まで打ち込んだらBREAKしてインサートモードを抜ける。ここで

E>TI

とすると，リストb)のようなリストが表示される。この状態ではBASICのようにカーソルを動かして自由に修正することができるので，間違いがあれば直す。修正が終わったら，またBREAKする。

次に

E>A

としてエディタモードを抜け，アセンブラモードに戻り，

A//

と打ち込むとアセンブルが行われ，本文のリストIが画面に表示される。エラーが起きたら，またエディタモードに入って修正することになる。

以上がZEDAを使ったソースの作成，アセンブルの一般的な手順だ。今後リストはリストIの形式で掲載される。なお，上の例でもわかるように，「このソースリストを打ち込みましょう」と書いてあったら，リストIのうちリストa)に該当する部分だけを打ち込むことになる。

リストa)

```
        ORG     8000H
;
#PRINT  EQU     1FF4H
;
        LD      A,90
        CALL    #PRINT
        RET
```

リストb)

```
1         ORG     8000H
2 ;
3 #PRINT  EQU     1FF4H
4 ;
5         LD      A,90
6         CALL    #PRINT
7         RET
```

かに」行ってしまった。

次に，0から1を引くとどうなるのだろうか。これも試してみよう。6行目から8行目を

```
    LD      A,0
    LD      B,1
    SUB     B
```

と変更してアセンブル，実行してみると，結果は

FF

となる。足し算の場合がわかれば，この結果の意味するものもピンとくるに違いない。0からでは1が引けないから，Z80は，'本当はないはずの上の桁から1を借りてきて，100_H-01_Hのつもりで計算する。結果は言うまでもなくFF_Hとなる。確かにFF_Hに01_Hを足すと00_Hになるのだから，00_Hから01_Hを引けばFF_Hになってもおかしくはない。

ここで，ひとつ面白いことがわかる。ある数Nのマイナスは0－Nと等しいはずだから，00_Hから01_Hを引いた結果のFF_Hは「マイナス1」だというようにも解釈できるだろう。だからこそ，FF_Hと01_Hを足すと00_Hになったという見方もできる。次の節ではマイナスの数の扱いについて，さらに突っ込んだ話をしてみよう。

255＝－1？

僕はA～Lのレジスタには0～255の値を格納することができると言った。ところが，FF_Hはどうやら－1として扱ってもよいようだし，さらに$FE_H=-2$, $FD_H=-3$というようにマイナスの数らしいものが現れてきた。この矛盾からは，僕があっさり引き下がることで解決される。

そう，FF_Hは－1と言っていい。

より正確には，プログラムのなかで－1として扱われていれば－1だし，255だとして扱われていれば255だと言える。場合によっては，どちらにも解釈できるんだ。一般にコンピュータでは負の数は「2の補数表現」という形式で表す。ある数の2の補数は，

1) まず，その数を2進数で書き表す
2) 1桁ずつ見て，1であれば0に，0であれば1に反転する
3) 得られた結果に1を足す

という手順で求められる。たとえば－2は

1) $2=00000010_B$
2) 0と1を反転すると$11111101_B=FD_H$
3) 1を足すとFE_H

となる。同じ手順をFE_Hに施すと，ちゃんと2に戻ることを確認してほしい。

こう考えていくと，ある数をプラスとみなすかマイナスとみなすかは，その人次第，プログラム次第ということになる。それではなにかと不便なので，マイナスの数を扱うとき，普通は2進数で表したときのいちばん左の桁（最上位ビット）が1であれば負の数，0であれば正の数とみなすことになっている。8ビットでは

10000000_B～01111111_B

10進数に直すと

－128～127

の範囲の数を表現できるわけだ。この表現は今後もちょくちょく出てくることになるから，頭の隅にでも入れておいてもらおう。

キャリフラグ

リスト2を使った実験で，255に1を足すと0になることがわかった。本当は繰り上がりがあったわけだが，レジスタの大きさの制限から，繰り上がった分はどこかに消えてしまっている，そう説明した。実は繰り上がった1ビット（2進1桁）は消えたのではなく，「キャリフラグ」に保管されている。だから，ADD命令で足し算をしたあとでキャリフラグを調べれば，桁上がりがあったかどうかを知ることもできる。

図1をもう一度見てほしい。Aレジスタの隣には，Fというレジスタがある。このレジスタは，これまでに話してきたAなどのレジスタとは違った性格のもので，1ビット1ビットが独立した意味を持っている。というより，1ビットずつ独立したものをいくつもまとめて，見かけ上8ビットレジスタの形にしたと考えてもらいたい。キャリフラグは，このF（フラグ）レジスタのなかの1ビットに付けられた名前だ。

いままではレジスタは「変数のようなもの」であって，値を格納して演算に使ったりしたわけだが，フラグレジスタはその性格のため，値を代入して保存しておくような用途には使えない。キャリフラグの例でもわかるように，演算結果によって自動的に値が変わってしまうからだ。文字どおり，フラグの集合でしかない。そのため

```
    LD      F,10
```

とか

```
    LD      F,A
```

のような命令はない。繰り返しになるが，フラグレジスタは1ビットごとに別々の意味があるのだから，全体（8ビット）をまとめて扱うのは意味のないことだ。

さて，キャリフラグのお陰で，足し算のときの桁上がりを知ることができるようになった。では，SUB命令で引き算をしたときの桁借りを知る方法もありそうだ。実はこのときもキャリフラグが使われ，借りが生じると，キャリフラグは1になる。

ADD命令のときは「繰り上がった1ビットがそのままキャリフラグに入る」と考えても，「繰り上がりが起こったことを表すために」キャリが1になったと考えてもよかった。けれども，引き算の場合は「借りが生じたことを表すために」キャリが1になるとしか考えようがない。ここらあたりで，「フラグ」というものの性質が少しはっきりしてくるように思う。

また，AND, OR, XORの各命令を実行すると「キャリフラグは常に0になる」。この場合，桁上がりも借りも生じないから，キャリが0になると考えてもらっていいだろう。

ここまでをまとめておくと，

1) ADD命令実行後，桁上がりが生じればキャリは1，生じなければ0
2) SUB命令実行後，借りが生じればキャリは1，生じなければ0
3) ADD, OR, XOR命令実行後はキャリ

16進数

我々が普段使っている10進数では，0～9の10種類の数字を使って数を表し，1桁で表せる最大の数である9に1を足すと繰り上がって10になる。コンピュータの世界でよく使われる16進数では，0～9とA～Fの16種類の文字を使って数を表し，やはり1桁で表せる最大の数F_Hに1を足すと，繰り上がって10_Hになる。ここで右下の小さな“H”は16進数で表記されていることを表している。

なお，ZEDAで16進数を記述するときは

$1234

のように頭に“$”を付けるか

1234H

のように末尾に“H”を付ける。

表に10進数と16進数，そしてその次によく使われる2進数の関係をまとめておく。

表1

10進数	16進数	2進数
0	0	0
1	1	1
2	2	10
3	3	11
4	4	100
5	5	101
6	6	110
7	7	111
8	8	1000
9	9	1001
10	A	1010
11	B	1011
12	C	1100
13	D	1101
14	E	1110
15	F	1111
16	10	10000
32	20	100000
255	FF	11111111

は常に0
となる。

ゼロフラグ

キャリフラグに続いて，もうひとつのフラグを紹介しておこう。「ゼロフラグ」という名のこのフラグは，名前が示すとおり演算の結果が0になったかどうかを表す。

たとえば，

```
LD      A,0FFH
ADD     A,1
```

のようにAレジスタにFFHを入れておいて1を加えると，結果は00Hなのでゼロフラグが1になる。同様に

```
LD      A,5
SUB     5
```

と，引き算の結果00Hになる場合もゼロフラグが立つ。AND, OR, XORの各命令を使ったときも，結果が0であればゼロフラグは1，そうでなければ0になる。

ここで，またまたサンプルを動かして遊んでもらうことにする。リスト3はAレジスタに適当な数を入れておいて（8行），足し算をしてみて（9行），結果の値（16進）とキャリフラグ，ゼロフラグの状態を表示するプログラムだ。結果は

Aレジスタ　キャリ　ゼロ

の形式で

46　0　0

のように表示される。例によって，8行目の値と9行目の演算命令を変えていろいろ試してみること。

10行目以降では，まだ説明していない命令をビシバシ使っているが，気にせずに実行結果だけに注目してほしい。

ところで，先ほどからなにげなく

```
SUB     1
```

のように直接値を記述する形式（イミディエイトアドレッシングというんだったね）を使っていることに気づいているだろうか。リスト2ではわざわざ

```
LD      A,なんとか
LD      B,かんとか
ADD     A,B
```

のように書いていたが，イミディエイトアドレッシングを使って

```
LD      A,なんとか
ADD     A,かんとか
```

と書いても同じ結果を得ることができる。

演算命令で使えるアドレッシングモードはほかにもあるのだが，いまのところレジスタ直接とイミディエイトだけを使っていくことにする。

無条件分岐

特に説明しなかったし，誰も気にはしなかったと思うのだがマシン語プログラムは（ソースで見ると）上から順に実行される。このあたりはBASICなんかの言語となんら変わらない。BASICでは処理の流れを変えたいときにはGOTO文を使った。マシン語にもそれに相当する命令，JPがある。

JPはJumPの略で，指定のアドレスに処理を移す（分岐する）命令だ。リスト4を見てもらおう（慌てて実行しないように）。このプログラムはJP命令を使った無限ループの例で，実行するといつまでも同じところをグルグル回るから，S-OSに戻るにはリセットスイッチのお世話にならなくてはならない。害のない「暴走プログラム」だ。

JPの後ろには

```
JP      8000H
```

のように分岐先アドレスを書くが，普通はリスト4でやって見せたように

```
JP      LOOP
```

とラベルを使う。ラベルを使うのは，プログラムの読みやすさを向上させる以外に，次のような必然性がある。

マシン語プログラムでは，分岐先はアドレスで指定しなければならない。すでにリスト1などで示したように，分岐先のアドレスが最初からわかっていれば，アドレスを直接記述することもできた。ところが，リスト4のようにプログラム内部に分岐する場合は，アセンブルしてみるまで分岐先アドレスがわからない。

そこで，「ここに分岐したい」というところにラベルという目印を付けておいて，JPやCALLなどの分岐命令ではこのラベルを使って分岐先を記述するようにする。こうしておけば，アセンブルするときにアセンブラが勝手にラベルのアドレスを計算してくれるんだ。アセンブラを使う利点はこんなところにもある。

リスト4を見てのとおり，ラベルは行頭に置くことで定義される。ラベルに使える文字はアセンブラによって違うが，ZEDAでは「数字で始まらず，スペース，カンマ，コロン，セミコロンを含まない任意の文字列」が使える。また，アセンブラによっては「ラベルは何文字まで」という制限があるが，ZEDAでは1行に収まる限り何文字でも許される。

また，説明が遅れてしまったが，アセンブラは行頭にスペースがないとラベルとみなすことになっている。だから，LDやCALLなどの命令を使うときには，直前にいくつかのスペースを入れなければならない。スペースの数に決まりはないが，プログラムを読みやすくするために，行頭を8文字落とすのが一般的な習慣となっている。

なお，リスト4ではラベルの直後にコロンをひとつ付けている。これも「習慣」で，ZEDAではコロンを付けなくてもいいのだが，なんとなく見栄えがよいような気がして，僕はいつも付けることにしている。

条件分岐

プログラムでは，ある条件に応じて処理

リスト3

```
8000                 1          ORG     8000H
8000                 2 ;
8000                 3 #PRINT   EQU     1FF4H
8000                 4 #PRNTS   EQU     1FF1H
8000                 5 #PRTHX   EQU     1FC1H
8000                 6 #LTNL    EQU     1FEEH
8000                 7 ;
8000 3E 12           8          LD      A,12H
8002 C6 34           9          ADD     A,34H
8004 F5             10          PUSH    AF
8005 CD C1 1F       11          CALL    #PRTHX
8008 CD F1 1F       12          CALL    #PRNTS
800B F1             13          POP     AF
800C F5             14          PUSH    AF
800D 3E 30          15          LD      A,30H
800F CE 00          16          ADC     A,0
8011 CD F4 1F       17          CALL    #PRINT
8014 CD F1 1F       18          CALL    #PRNTS
8017 F1             19          POP     AF
8018 3E 30          20          LD      A,30H
801A 20 01          21          JR      NZ,SKP
801C 3C             22          INC     A
801D CD F4 1F       23 SKP:     CALL    #PRINT
8020 CD EE 1F       24          CALL    #LTNL
8023 C9             25          RET
```

リスト4

```
8000                 1          ORG     8000H
8000                 2 ;
8000                 3 LOOP:
8000 C3 00 80        4          JP      LOOP
```

を振り分けることがある。BASICでなら

IF~THEN~

というような構文がこれにあたり，Z80にも同じような機能を果たす命令がある。といってもZ80にはIF~THENのようなオールマイティな命令はなく

IF~GOTO~
IF~GOSUB~
IF~RETURN

に相当する命令がバラバラに存在するだけだ。それぞれすでに出てきた

JP
CALL
RET

の各命令を使う。書式は

```
JP     条件，分岐先アドレス
CALL   条件，分岐先アドレス
RET    条件
```

となる。

さて，なにをもって条件とするかだが，マシン語では「フラグの状態」を条件として利用する。つまり，「キャリフラグが1なら指定のアドレスへ分岐する」，「ゼロフラグが0ならサブルーチンを呼び出す」といった使い方をするわけだ。具体的な例を挙げると

```
JP     C, ERROR
CALL   NZ, TEST
RET    NC
```

のように書く。CとかNZとかは「フラグの状態を表す記号」で

C：キャリフラグが1ならば
NC：キャリフラグが0ならば
Z：ゼロフラグが1ならば
NZ：ゼロフラグが0ならば

の意味になる。用語を持ち出しておくと，NC, NZはそれぞれ「ノンキャリ」，「ノンゼロ」と読む。

ここで，読者にクイズを出そう。「Aレジスタの値とBレジスタの値を比べてみて，等しければ指定アドレスへ分岐する」にはどうしたらよいだろうか。仮に分岐先アドレスは"TEST"というラベルで表すことにしよう。いままでに紹介した命令だけで実現できるから，しばらく考えてみてほしい。

たぶん，もうクイズの答えはわかっていると思うが，念のため確認しておこう。

まず，最後の「等しければ指定アドレスへ分岐する」処理は

```
JP     条件, TEST
```

という形になるのは間違いない。

```
JP     A=B, TEST
```

なんて書き方が許されれば，話はこれで終わりだが，あいにくJP命令で使える条件は「フラグの状態」しかないんだった。そこで，JPの前に別の命令でフラグをセットしておき，そのフラグの状態によって分岐するようにしなければならない。どの命令を使うかだが，SUB命令を利用するのがよさそうだ。

もしAレジスタとBレジスタが等しければ，

```
SUB    B
```

を実行した結果は0となり，ゼロフラグが立つ。そのあとで

```
JP     Z, TEST
```

とすれば目的を達成することができる。つまりクイズの答えは

```
SUB    B
JP     Z, TEST
```

となる。この組み合わせで，BASICでいう

IF A=B GOTO~

に相当する処理が行えるわけだ。逆に「AとBが等しくなければ分岐する」には

```
SUB    B
JP     NZ, TEST
```

とする。

では，クイズ第2弾。「AとBを比べて，Aのほうが小さければ分岐する」にはどうすればよいか。

今度はゼロフラグではなく，キャリフラグを使えばいい。AレジスタからBレジスタを引いてみて，「借り」が生じればBのほうが大きかったことがわかる。そこでクイズ2の答えは

```
SUB    B
JP     C, TEST
```

となる。これも逆に「AがB以上であれば分岐する」ようにしたければ

```
SUB    B
JP     NC, TEST
```

とする。

ここで，

```
SUB    B
```

実行後のフラグの変化を，元のAレジスタとBレジスタの値の大小関係別にまとめておくと

A<B　キャリ，ノンゼロ
A=B　ノンキャリ，ゼロ
A>B　ノンキャリ，ノンゼロ

のようになる。

では，クイズ第3弾。やはりAレジスタとBレジスタを比較して，AがB「以下」ならば分岐させるにはどうすればよいだろう。

答えは次のようになる。

```
SUB    B
JP     Z, TEST
JP     C, TEST
```

JP命令を2つ並べるのがポイントで，最初にAとBが等しければ先に飛ばしてしまい，続いてAがBより小さい場合をチェックしている。

さらにクイズ第4弾。今度のはちょっと難しい。AがB「より大きい」ならば"TEST"へ分岐するようなプログラムを考えてみてほしい。もちろん

```
JP     NC, TEST
```

ではA=Bの場合も分岐してしまうから，うまくない。

きっと悩んでいるだろうから，答えを教えよう。こうなる。

```
      SUB    B
      JP     Z, SKIP
      JP     NC,TEST
SKIP: ~
```

どういうことかというと，A=Bならば最初の

```
JP     Z, SKIP
```

で，"SKIP"というラベルに分岐する。見てのとおり"SKIP"へ飛んでくると，もう"TEST"へのJP命令があるところを通らなくなってしまう。で，

```
JP     NC, TEST
```

は「A>=Bならば~」の意味だが，すでにA=Bではないことが保証されているから，実質的に「A>Bならば~」の意味になるわけだ。

では，最後のクイズ。「A>Bならばサブルーチンをコールするにはどうしたらよいか」

一見，JPがCALLに変わっただけでクイズ3と同じだと思うかもしれない。そこで，まず考えるのは

```
SUB    B
CALL   Z, TEST
CALL   C, TEST
```

のようなプログラムだろうが，残念ながら

アドレス

コンピュータのメモリは"バイト"という単位の小さな区切りに分けられている。それぞれには「番地」とか「アドレス」と呼ばれる0から始まる通し番号が付けられており，「8000H番地」のようにして参照される。仮にメモリを配列にたとえると，アドレスは添え字にあたるといえる。

Z80のメモリ空間は64Kバイト（1Kバイト=2^{10}バイト=1024バイト）だから，0000H番地からFFFFH番地までが存在することになる。

また，1バイトは「0か1かを表すもの」を8つまとめたもので，「0か1か~」のことを"ビット"という。感覚的には1ビットは2進数の1桁に相当し，1バイト=8ビットだから，2進数8桁の範囲の値を格納できることになる。

間違いだ。仮にAとBが等しかった場合を考えてみよう。

A＝BならZフラグが立つ。そこで，

```
CALL    Z,TEST
```

でサブルーチンへ分岐する。めでたしめでたしかと思うと，サブルーチンを呼び出したのだから，サブルーチンのRET命令で

```
CALL    C,TEST
```

があるところに戻ってくる。まだ救いはあって，A＝BだったのだからノンキャリなのでこのZ行は素通りしてくれそうに見える。けれども，「もしかするとサブルーチン側でもADDやSUBなんかの演算命令を使っていて，フラグが変化しているかもしれない」。運悪くキャリが立って帰ってくると

```
CALL    C,TEST
```

で，もう一度サブルーチンを呼び出してしまうことになる。

というわけで不正解。正しい答えは来月発表しよう。それまでゆっくり考えてみてほしい。

キャラクタ募集のお知らせ

マシン語の話はここまでにして，最後にお待ちかねのゲームの話をしておこう。

横スクロールシューティングといってもたいしたものを作るわけじゃない。グラフィックなどは使わず，40×25のテキスト画面だけですませる。当然，滑らかな動きなんてのからはほど遠いものになるだろう。

また，データ量を極力押さえたいので，敵キャラの種類も片手に足りるぐらい（たぶん2種類か3種類）にする。背景も2画面分だけ用意して繰り返すつもりだ。面もひとつあればいいだろう。タイトルも付けなければ，BGMや効果音も考えない。もちろん，アイテムの類を用意するつもりもない。

テキストでゲームが作れればグラフィックでも作れる。2種類の敵キャラが動かせるようになれば，20種類だろうが30種類だろうが動かせる。2画面の繰り返しで背景がスクロールできるようになれば，データを増やせばいくらでも長くできる。1面ができれば，2面目もできる。ゲーム本体ができればタイトルを付けるくらい簡単だ。

そんなノリで押してみようと思う。

で，僕はキャラクタを作るのがおっくうなので，キャラクタパターンは読者の方それぞれに用意してもらうつもりでいる。以下に規格をまとめるから，この線に沿って作っておいてほしい。

1) キャラクタの大きさは2×2文字
2) 背景は3×3文字の大きさのパーツをいくつか作って組み合わせる
3) 弾は1文字の大きさで，自機が前方にまっすぐ撃つ弾と，敵が自機を狙って撃つ弾の2種類を用意する
4) X1およびMZ-2500では8色の8×8ドットPCGを使う
5) MZ-700はキャラグラで8色でもいいし，古籏君方式の2画面重ね合わせ疑似36色でもいい
6) 背景は右から左へ流すので，自機は右向き，敵キャラは左向きとなる
7) 敵キャラの「どれかひとつ」は，2ないし4パターン程度のアニメーションにする予定もある
8) これもあくまで予定なのだが，彩りを添えるために6×6もしくは8×8程度の大きさのデカキャラがひとつあってもいいように思う
9) データのフォーマットはいまのところ特に指定しない。BASICで表示できるようにしておけば十分だ

そうそう。この際だからサンプル用のキャラクタを募集してしまおう。自信作ができたらどんどん送ってきてほしい。キャラクタパターンだけでなく，敵キャラなら移動/攻撃パターンなんかも付けてくれると嬉しいな。ほかにもなにか要望があれば（グラフィックを使えとか，背景が2画面分じゃ少ないなんて寝ぼけたのはダメだぞ）ありがたーく参考にさせてもらうから，そっちのほうもよろしく。

ふう，やっと終わった

くっそー。今月は思いっきり死んだ。Z80については知り尽くしているつもりだったのに，いざ，説明しようとしたら難しいのなんの。かなり説明過多になってしまった。

えっ？「説明不足の間違いじゃないか」って？ ちゃうちゃう。ほんとはもっとあっさりクールにまとめたかったんだ。ところが，ついついおせっかいして余計なことまで話してしまったので，混乱してしまった読者もいるかもしれない。

逆にいうと，ここに書いてあることを全部覚えようとする必要はないということだ。命令は使っていれば自然に覚えるものだし，用語に関してもそうだといえる。それよりも，いくつか挙げた小さなプログラム例を実際に走らせて，気のすむまであれこれ試してみてほしいと思う。

次回も引き続きZ80の命令解説をする。並行して実践向きの細かなテクニックなんかも紹介してみたい。

じゃ，来月また会おう。

疑似命令

アセンブリ言語には疑似命令というものがある。これはマシン語本来の命令ではなく，「アセンブラに対する指示」をする命令群だ。いくつかの疑似命令があるのだが，ここではよく使うORGとEQUの2つについてだけ説明しよう。

ORG命令はORiGinの略で

```
ORG    アドレス
```

の形式で書き，オブジェクトが「どの番地から始まるのか」を指定している。たとえば

```
ORG    9000H
```

とあれば，プログラムは9000H番地から始まるようにアセンブルされることになる。特にアブソリュートアセンブラではORG命令で指定されたアドレスからオブジェクトを直接生成するので，ORG命令を付け忘れたり，変なアドレスを指定したりすると，モニタやアセンブラを破壊する危険性があり，アセンブル時に暴走してしまうことさえある。

本稿ではオブジェクトは可能な限り8000H番地から始まるように統一する。だから，どのサンプルを見ても，先頭に

```
ORG    8000H
```

と書いてあるはずだ。また，オブジェクトが8000Hから置かれるのだから，実行するにはS-OS“SWORD”上から

```
#J8000
```

とするか，またはZEDAのアセンブラモードで

```
J8000
```

と入力することになる。

なお，Z80のプログラムは多くの場合，アセンブルのときに指定したアドレス以外では動作しない。だから，

```
ORG    8000H
```

としてアセンブルしたプログラムをいったんセーブしてから9000H番地にロードしたとしても実行することはできない。

次に，EQU疑似命令はEQUateの略で，「ラベルに定数を定義する」命令だ。

```
TEISUU    EQU    100
```

のような書式で使い，以後，100と書く代わりにTEISUUというラベルで記述できるようになり，プログラムを読みやすくすることができる。

注意しなければならないのは，ラベルは必ず行頭から書き始めなければならないことだ。上の例だと，“TEISUU”の前に余分なスペースを入れてはいけない。

また，疑似命令ではないのだが，プログラムの可読性を上げる目的で，アセンブリソースには注釈を入れることができる。具体的には，セミコロン以後から行末までは注釈とみなされる。

フラグ

「ある状態かどうかを0か1で表すもの」を“フラグ”という。BASICでも変数をひとつ用意して，普段は0にしておき，なにかの条件が整ったら1にする，といった使い方をするだろう。これもフラグの例といえる。

なお，フラグを文字どおり旗に見立てて，1にすることを「立てる」といい，0にすることを「倒す」ということもある。

Object oriented

実用(?)オブジェクト指向のゲームプログラミング――最終回

オブジェクト指向のゆくえ

Hamaguchi Isamu
浜口　勇

なんとかかんとかサンプルのスネークゲームも出来上がり，完成した証しとしてオブジェクトのダンプリストも掲載する。この連載もこれでおしまいとなるが，最後にいままでのまとめと今後に対する展望などを考えてみることにしよう。

オブジェクト指向の機能

連載を通してわかりにくいという意見が多かったのだが，これの大きな原因のひとつは前処理を行うためのフィルタの意味や働きと実際にアプリケーションを作るための方法（というか考え方）を同時進行で説明してきたためだという気がする。そこで，ここではclassm自身の機能に絞り込んで説明してみよう。

「オブジェクト指向が～」といままで説明してきたのだが，いったいどこからどこまでがオブジェクト指向なのかは漠然としてはっきりしない。ちょうどかつての構造化プログラミングのようなものだと考えればよいだろうか？　構造化に関してはダイクストラ先生の原点があるわけなのではっきりしているのだが，オブジェクト指向に関したバイブルといったものはどこにも存在しないように感じられる。Cにだって K&Rがあるのにだ。

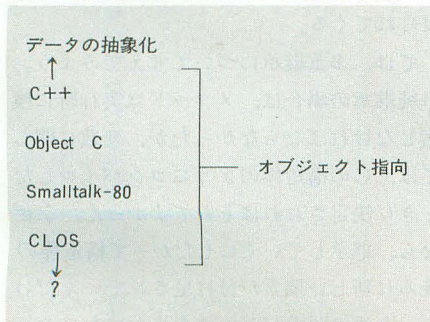
それでもかつてはSmalltalk＝オブジェクト指向という図式があったのでよかったのだが，今現在はそれもかなり怪しい。ちょっと考えただけでもC＋＋，Object C，CLOSなどなど思いつくぐらいで，『日経バイト』に載ったオブジェクト指向言語の特集には聞いたこともないような言語の名前がたくさん出てきていた。そのうちIBMのいうCOBOL構造化プログラミングのようにとんでもないオブジェクト指向も出てくる可能性もないとはいえない。

ここで，これらの言語の指し示すオブジェクト指向というものを表1にまとめてみた。

表1　さまざまなオブジェクト指向型言語

C＋＋とObject C

まず抽象データ型に近い（というかほとんどそのものということになっている）のがC＋＋。ちなみに『UNIX原典』(パーソナルメディア刊）という本では「Cにおけるデータ抽象化」というタイトルでC＋＋について解説している。一応，C＋＋の売りとしてオブジェクト指向をあげているところもあるが，結構怪しい面も多い。一般にこういったデータを抽象化して扱う機能は，Cでは構造体の拡張として，PASCALなんかではレコードに対する拡張として持たせるのが流行みたいになっている。

C＋＋の場合はかなり汎用的に作られているのでやり方によってはあとで述べるObject Cみたいな処理を行うことも可能である。ただし，Object Cのソースだって最終的にはCに落ちるのである。やろうと思えばできる，と，それ用に作られているとは大きな違いがあるということである。

それにしても，ただでさえ難しいといわれているCにさらに＋＋した言語ということで我々一般人の能力を超越している部分があるのも事実で，私にもよくわからない。

さて，次はObject Cだが，これはざっと見た限りではclassmともたいして変わらない。Object C自身はオブジェクト指向まっしぐらなのでC＋＋に比べるとずいぶんとわかりやすい。

変数に関しては明らかに，単なる構造体を使っているのだが，メソッドの呼び出しにどういった方法を使っているかははっきりしない（参考文献がよくなかった気もするが）。しかし，いろいろな解説の記事を合わせて考えると，classmと同じ方法だろうと考えられる。つまり関数へのポインタを使うわけである。安易な方法ではあるが，これ以上複雑な方法を取ると，リアルタイムな応用に使用できなくなってしまう。

それでは，Cの構造体をどうやって使うかについて説明していこう。

まず，スネークゲームのクラスobjectがCではどのように表される？　これについて考えてみると，次のような定義が作られるだろう。

```
struct    mateclass {
          struct mateclass *mclass;
          struct c_method *method;
}m_object={&m_object,&object_cmethod};
struct    class {
          struct mateclass *mclass;
          struct i_method *method;
          unsigned int memsiz;
} object={&m_object,&object_imeth
od,sizeof(struct instance)};
struct    instance {
          struct class *class;
} i_object;
struct    c_method {
          (*alloc)( );
          (*new)( );
} object_cmethod={&object_alloc(),
&object_new};
struct    i_method {
          (*free)();
```

```
        (*freeobj)();
}object_imethod={&object_free(),
  &object_freeobj()};
```

ざっとこういった感じである。これでメソッドのnewを呼び出すには，クラスへのポインタを使って，

```
  ob=(*(object.method->new))(&object);
```

というふうに行えばよいのでかなり楽だ。

変数へのアクセスも，

```
  i=cl->memsiz;
```

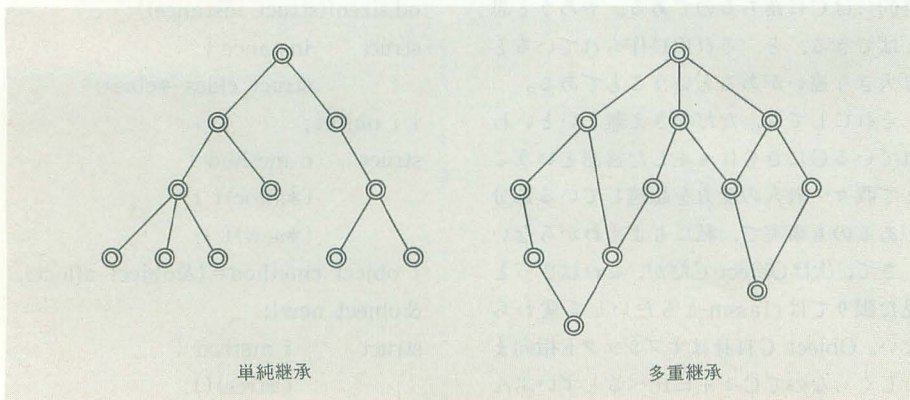
のように行えばよいのである。だからZ80などのそれほど機能が高くないCPUでも，LSI-CやMSX-Cのように実用性の高いCコンパイラを使うことによって，容易に実現できる。

実際にclassmをC用に拡張する場合に問題になるのは，いまアクセスしているオブジェクトを指すためのポインタをどうするかという問題である。

classmではBCレジスタに割り振っていたのだが，Cでは明示的にどのレジスタにレジスタ変数を割り振る，といったことができないので，レジスタの代わりに大域変数に割り振るといった処置が必要になる。しかし私自身としては，前記の例でもそうなっているように，関数の第1引数を必ずオブジェクトへのポインタとして割り振るという方法を使っている。そうすれば受け側のメソッドでは，

```
  object_new(cl)
  struct class *cl;
  {
```

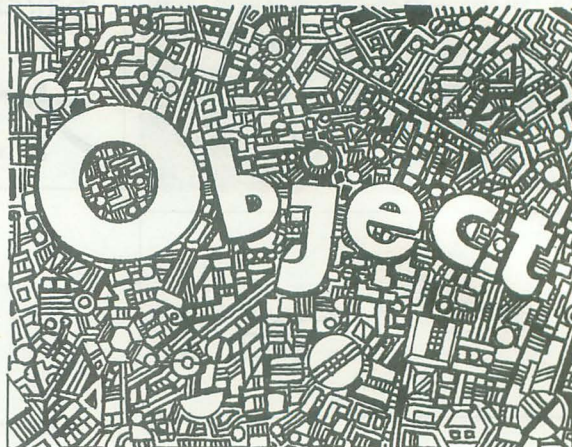
というかたちで受ければよいことになる。

もしもこれを大域変数で行おうとすれば，同じ変数の型がオブジェクトごとに異なるという処理を行わなければならなくなり面倒である。幸い，LSI-Cなどでは第1引数をHLレジスタに割り振っていてメソッドがアセンブラで書かれていてもそれほど処理に変化はない。しかし，通常のCの場合はスタック上に積まれるために処理を少し変えてやらなければならなくなるだろう。すべてをアセンブラで行うのではなくて，このようにCを中心にして細かいところをアセンブラで処理するというふうにすると開発もかなり容易になるし，プログラム全体の見通しもよくなる。

Smalltalk

さて，その次にくるのがSmalltalkである。これはSmalltalk-80に限ってもバージョンアップがなされていて，ひとつではない。つまり多重継承が可能になる前と後では大きく異なるということがいえるのである。

ひとつのクラスに対してその親（スーパークラス）がひとつなのを単純継承，複数あるのを多重継承といって分けている（図1）。

単純継承のSmalltalkというのは，classmなんかと比べてもたいして変わらないじゃないかと思われるかもしれないが大きく異なる点が2つある。

まずひとつ目は，呼び出されるメソッドが決定実行時にされるということである。たとえばclassmではどのオブジェクトを呼ぶかはアセンブルする前に決定しておいて，定義してやらなければならない。しかし，Smalltalkの場合には実行時に呼び出すメソッドの名前を基に，メソッドのテーブルを検索してメソッドを探し出す。つまり呼び出し側は，どんなオブジェクトにメッセージを送るか，ということをまったく知らなくてもよいことになる。

さて，この差はどれくらい大きいのだろうか？　当然，Smalltalkのほうが自由度が高い（オブジェクトさえわかれば，それがどんなクラスに所属していようと，メッセージを送れる）。しかし，メソッドの検索を行う限りリアルタイムゲームには使用できないというのは，事実である。つまり処理時間に幅ができてくるからだ。

オブジェクト指向で疑問なのはこのあたりである。メソッドを実行時に決定しない言語はオブジェクト指向と言えないのだろうか？　そして，それは言語の処理能力にどのようにかかわってくるのだろうか？　誰か，知っていたら教えてほしいものである。

さて2つ目の差だが，Smalltalkという言語は一度作成されたオブジェクトを解放する命令を持たないということである。つまり，作りっぱなしなわけで，あとはシステムが管理してくれるために，ユーザーは間違って解放したオブジェクトに対してアクセスするといったミスを犯す心配がない。これは逆にいうとLisp同様死ぬほどメモリを使うということで，用途はどうしても限られてくる。

では，多重継承について考えてみよう。単純継承の場合は，メソッドは実行時に検索しなければならなかったが，変数に関しては，Cの構造体のようにコンパイルしたときに決定されればそれでよかった。なぜなら，継承していくにしたがって構造体の後ろに新しい要素が付け足されていくだけで，全体の順番が変化することはないから

図1　単純継承と多重継承

だ。ところが多重継承ではそうはいかない。ある変数がメモリ上のどこにあるかは，一定ではないために，先頭からの順番とかではアクセスできないのである。

たとえばクラスAとBを多重継承したCを考えると，

```
struct A{
        int     aaa;
        int     bbb;
};
struct B{
        int     ccc;
        int     ddd;
};
struct C{
        int     aaa;
        int     bbb;
        int     ccc;
        int     ddd;
};
```

クラスCのメソッドの中では，

i=cl－>ccc;

という式は正しいが，クラスBのメソッドの中で，同じ式を使うと，cccは変数の最初の要素を指すというかたちにコンパイルされてしまっているため，実際にはaaaに対してアクセスしてしまう結果となる。そのため，変数の名前を使って毎回変数を探してやらなければならない。つまり，多重継承の場合の変数は実行時にならなければ決定しないということなのだ。

たとえばBASICのようなインタプリタでも，変数に1回でもアクセスすれば，その絶対番地が決定するので，中間コードの中にその番地を格納して処理の高速化をはかるのだが，この場合は変数をアクセスするたびに変数の番地を必ず求めなければ正しい処理は望めない。そういう意味では，当然BASICよりも処理スピードは遅くなるだろう。そして，それにも増してさらに処理時間を推定するのが困難になる。つまりリアルタイム処理向けでなくなるわけである。

未知数のCLOS

それでは最後に，複数のオブジェクトに対してメッセージを送れるというCLOSを見てみよう。

これは『bit』の記事で読んだだけなのであまり詳しいことは知らないが一度にひとつのオブジェクトにしかメッセージを送れないのは不自由だということらしい。もし一度に複数のオブジェクトにメッセージを送れるならば，いままで行ってきたひとつのオブジェクトに対してだけメッセージを送るというのも，その特別な場合として使用できるということになるわけである。これは，一見して普通の関数とあまり変わらなくなってしまう。

たとえばオブジェクトAとBにCというメッセージを送るというのをC言語風に書くと，

C（A，B）;

となって，どこがオブジェクト指向なのかわからない。しかし，実際にはA，Bがどのクラスに属したオブジェクトかによって呼び出されるメソッドCは毎回異なることになるわけである。

たとえば，情けない例で申し訳ないのだが，

printf（"%d", n);

とやっても，

printf（fd,"%d", n);

とやっても，それぞれ呼び出されるメソッドが違うのだからよいことになる。なぜなら，最初の引数のクラスによってメソッドが選択できるからだ（これはべつに"%d" printf：nとfd printf："%d"nでもかまわないのだけどもね）。ここまでくるとProlog的になってくるような気がするが，そうなるとどうなるか？　ということはまだよくわからないみたいだ。

ROM化について

私の場合ビデオゲームのプログラムがテーマであり，ROM化は避けられない問題だ。実際にはclassmの場合，インスタンス変数のみRAM上に展開するようにしている。これはちょうど，LSI-Cなんかが取っている方法と同じといえるだろう。LSI-Cでは初期化された変数や構造体はすべてdseg（ROM上）にくるようになっている。

classmでは初期化された構造体＝クラス変数ということで，クラス変数には参照はできるけれども，変更ができないという制限が付くことになった，なにしろROM上にあるわけだからね。つまりLSI-Cとかを使えばCで作ったプログラムでもROM化できるわけで，Cでオブジェクト指向をやりたい向きには最適かもしれない。

バグを飼い馴らす

この連載で心残りなのは，もう少し大きめのプログラムを紹介したかったということだ。また，Cとかを使わないとなかなか実用的な線はクリアできないというのも事実で，やはり16ビット機向けのものになってしまうのかもしれない（CPUの機能というよりも，単にCが普及しているかしていないか，という点において）。

ところで，個人的にオブジェクト指向のどこが良いかというと，実はバグが出ると

オブジェクト指向に関する参考文献

梅村恭司，「Smalltalk-80入門」，サイエンス社
安いし，薄いし，内容もよい，という牛丼のような本。結局これがいちばん役にたったという点で，言語的に攻めたい人にはお勧めですね。

鈴木則久編，「オブジェクト指向　解説とWOOC'85からの論文」，共立出版
Smalltalk以外のオブジェクト指向についてもいろいろと載っており，バラエティに富んでいるので読んでいて面白い。ただ，論文というだけあって，すぐに役立つようなものではない。

館野昌一，及川一成，田制貴俊，「基礎からのSmalltalk-80」，工学社
こういった本は，実際にSmalltalk-80を使える環境にある人が読むと身につくのだろうが，一般のパソコンユーザーには必ずしも勧められないかもしれない。

特集「オブジェクト指向プログラミング」，インターフェース，　1987.1，CQ出版
これからはオブジェクト指向の時代だ，頑張るぞー，といったやる気がわいてくる特集。ただしこれも，すぐに役に立つというものではない。オブジェクト指向の概念はなんとかなったが，どうすればアプリケーションができるだろう？　といったところまでだったのが残念。

上谷晃弘，「統合化プログラミング環境――Smalltalk-80とinterlisp-D――」，丸善
筆者連がDTPの富士ゼロックスだけあってテクニカルイラストやレイアウトはかくのごとくあるべきだ，というお手本のような本。特に内容を印象的に見せるテクニカルイラストには，感心させられる。

いうことなのである。それも，プログラムが実際に動き出してから予想外のバグが出るのである。作ってもいない機能が半分崩壊しながらも動いたりするのは楽しいし，かえってイマジネーションを掻き立てられることになる。

ちょっとあぶない気もするが，そういった副作用を飼い馴らすことができればゲームプログラミングの面白さも増すことだろう。たとえば，アドベンチャーゲーム記述言語として期待する向きが多いのも，そうした副作用のせいだと思う。

半分壊れながらも，動き続ける融通というのはコンピュータがなかなか持てない能力のひとつではないだろうか。

リスト1　スネークゲームのダンプリスト(X1/X1turbo)

```
0100 00 00 00 F3 16 00 1E 12 : 39
0108 21 51 01 01 00 18 ED 51 : CA
0110 03 7E ED 79 23 14 1D C2 : FD
0118 0B 01 01 03 1A 3E 0D ED : 62
0120 79 11 02 08 ED 53 00 08 : DC
0128 01 00 04 21 20 00 19 EB : 4A
0130 73 23 72 0B 78 B1 C2 2B : 29
0138 01 C3 67 01 D5 E5 60 69 : AF
0140 5E 23 56 13 13 EB 5E 23 : 69
0148 56 E1 19 5E 23 56 EB D1 : E3
0150 E9 37 28 2D 34 1F 02 19 : E3
0158 1C 00 07 00 00 00 00 00 : 23
0160 00 00 00 63 01 67 01 01 : CD
0168 A3 06 21 00 00 CD 3C 01 : D4
0170 01 6C 06 21 00 00 CD 3C : 9D
0178 01 01 1F 03 21 00 00 CD : 12
---------------------------------
SUM: 7B 75 B2 CA 39 E7 C5 B1 1E88

0180 3C 01 01 05 88 21 08 00 : F4
0188 CD 3C 01 C9 8C 01 90 01 : F1
0190 AB 01 9E 01 C5 01 B8 01 : CA
0198 8C 01 94 01 02 00 2A 00 : 4E
01A0 08 5E 23 56 2B ED 53 00 : 4A
01A8 08 EB C9 21 02 00 CD 3C : E8
01B0 01 EB 71 23 70 2B EB C9 : CF
01B8 60 69 ED 5B 00 08 22 00 : 3B
01C0 08 73 23 72 C9 21 02 00 : FC
01C8 CD 3C 01 C9 CC 01 D0 01 : 71
01D0 E4 01 9E 01 2D 02 B8 01 : 6C
01D8 6A 02 0C 02 F7 01 CC 01 : 3F
01E0 D4 01 06 00 CD AB 01 21 : 75
01E8 02 00 19 AF 77 23 77 21 : FC
01F0 04 00 19 77 23 77 C9 21 : 18
01F8 04 00 09 C5 4E 23 46 79 : 02
---------------------------------
SUM: B2 8F 8D EE E6 D0 84 E6 3050

0200 B0 CA 0A 02 21 08 00 CD : 7C
0208 3C 01 C1 C9 21 02 00 09 : F3
0210 73 23 72 EB 5E 71 23 56 : 3B
0218 70 21 04 00 09 73 23 72 : A6
0220 7B B2 CA 2C 02 21 02 00 : 48
0228 19 71 23 70 C9 21 04 00 : 0B
0230 09 5E 23 56 D5 21 02 00 : D8
0238 09 5E 23 56 EB E5 5E 23 : 31
0240 56 EB A7 ED 42 E1 CA 4D : 0F
0248 02 11 04 00 19 D1 73 23 : 97
0250 72 7B B2 CA 66 02 21 02 : F4
0258 00 19 E5 21 02 00 09 5E : 88
0260 23 56 E1 73 23 72 CD C5 : F4
0268 01 C9 21 04 00 09 5E 23 : 79
0270 56 C9 72 02 76 02 8A 02 : 97
0278 9E 01 AE 02 B8 01 6A 02 : 74
---------------------------------
SUM: 57 67 D8 51 48 68 32 7D F59A

0280 0C 02 96 02 72 02 7A 02 : 96
0288 08 00 CD E4 01 21 06 00 : E1
0290 19 AF 77 23 77 C9 21 06 : C9
0298 00 09 C5 4E 23 46 79 B0 : AE
02A0 CA A9 02 21 08 00 CD 3C : A7
02A8 01 C1 CD F7 01 C9 21 06 : 77
02B0 00 09 C5 4E 23 46 79 B0 : AE
02B8 CA CC 02 21 04 00 CD 3C : C6
02C0 01 D5 21 00 00 CD 3C 01 : 01
02C8 C1 C3 B6 02 C1 CD 2D 02 : F9
02D0 C9 D1 02 D5 02 8A 02 9E : 9D
02D8 01 AE 02 B8 01 6A 02 0C : E2
02E0 02 96 02 EB 02 D1 02 D9 : 33
02E8 02 08 00 21 06 00 09 C5 : FF
02F0 4E 23 46 79 B0 CA 09 03 : B6
02F8 21 04 00 CD 3C 01 D5 21 : 25
---------------------------------
SUM: C1 D5 58 BF F5 6B A4 55 7C1E

0300 0A 00 CD 3C 01 C1 C3 F3 : 8B
0308 02 C1 C9 0B 03 0F 03 2A : D6
0310 03 26 03 AE 02 A8 03 6A : F1
0318 02 0C 02 63 03 7C 03 0B : 00
0320 03 13 03 0B 00 01 11 05 : 3B
0328 88 C9 CD 8A 02 D5 C5 C5 : 09
0330 01 1C 88 21 04 00 CD 3C : D3
0338 01 C1 21 06 00 09 7E F5 : 65
0340 32 15 88 01 DB 04 21 04 : D4
0348 00 CD 3C 01 42 4B 11 0B : B3
0350 88 21 06 00 CD 3C 01 F1 : AA
0358 A7 CA 60 03 3D C3 3F 03 : 16
0360 C1 D1 C9 21 20 4E 2B 7D : 92
0368 B4 C2 66 03 CD 96 02 21 : 65
0370 06 00 09 5E 23 56 7B B2 : 13
0378 C2 63 03 C9 ED 5B 02 88 : C3
---------------------------------
SUM: 3C 6F 79 64 33 B6 09 68 F645

0380 21 09 00 19 7E FE 01 FA : BA
0388 A2 03 FE 27 F2 A2 03 21 : 82
0390 0A 00 19 7E FE 01 FA A2 : 3C
0398 03 FE 18 F2 A2 03 CD EB : 68
03A0 02 C9 3E FF 32 04 88 C9 : 8F
03A8 C9 A9 03 AD 03 8A 02 9E : 4F
03B0 01 C9 03 58 04 B8 01 6A : 4C
03B8 02 0C 02 96 02 E0 03 37 : C2
03C0 04 13 04 A9 03 B3 03 0B : 88
03C8 00 21 00 00 CD 3C 01 21 : 4C
03D0 09 00 19 3A 10 88 77 21 : 8C
03D8 0A 00 19 3A 11 88 77 C9 : 36
03E0 2A 02 88 A7 ED 42 CA 0F : 63
03E8 04 ED 5B 02 88 21 09 00 : 00
03F0 19 7E 21 09 00 09 BE C2 : 4A
03F8 0F 04 21 0A 00 19 7E 21 : F6
---------------------------------
SUM: 0B F6 D0 23 B1 4E 5A B8 C517

0400 0A 00 09 BE C2 0F 04 21 : C7
0408 08 00 09 7E 32 04 88 CD : 1A
0410 EB 02 C9 21 09 00 09 7E : 67
0418 32 1E 88 21 0A 00 09 7E : 8A
0420 32 1F 88 21 08 00 09 7E : 89
0428 32 20 88 C5 01 1C 88 21 : 65
0430 06 00 CD 3C 01 C1 C9 21 : BB
0438 09 00 09 7E 32 1E 88 21 : 89
0440 0A 00 09 7E 32 1F 88 3E : A8
0448 20 32 20 88 C5 01 1C 88 : 64
0450 21 06 00 CD 3C 01 C1 C9 : BB
0458 21 0C 00 CD 3C 01 CD AE : B2
0460 02 C9 62 04 66 04 8A 02 : 27
0468 9E 01 82 04 58 04 B8 01 : 3A
0470 6A 02 0C 02 A7 04 E0 03 : 08
0478 37 04 13 04 62 04 6C 04 : 28
---------------------------------
SUM: 4F 73 75 CC 79 40 40 12 43F1

0480 0D 00 CD C9 03 C5 D5 3A : 7A
0488 14 88 21 08 00 19 77 42 : 97
0490 4B 21 0E 00 CD 3C 01 D1 : 55
0498 C1 21 0B 00 19 D5 ED 5B : 23
04A0 12 88 73 23 72 D1 C9 CD : 09
04A8 96 02 21 0B 00 09 5E 23 : 4E
04B0 56 1B 72 2B 73 7B B2 C2 : 70
04B8 C0 04 21 00 00 CD 3C 01 : EF
04C0 C9 C1 04 C5 04 E1 04 9E : DA
04C8 01 08 05 58 04 B8 01 6A : 8D
04D0 02 0C 02 60 05 E0 03 37 : 8F
04D8 04 13 04 C1 04 CB 04 13 : C2
04E0 00 CD 8A 02 21 0E 00 19 : A1
04E8 D5 11 1E 00 73 23 72 D1 : DD
04F0 21 0C 00 19 D5 11 3C 00 : 68
04F8 73 23 72 D1 AF 21 10 00 : B9
---------------------------------
SUM: 24 68 57 54 F7 B8 19 97 B2D7

0500 19 77 21 11 00 19 77 C9 : 1B
0508 3A 15 88 6F 26 00 29 11 : A6
0510 3E 06 19 5E 23 56 EB 7E : 9D
0518 32 10 88 23 7E 32 11 88 : 36
0520 23 7E 32 16 88 23 7E 32 : 44
0528 17 88 23 7E 32 18 88 23 : 35
0530 7E 32 19 88 CD C9 03 3A : 24
0538 15 88 21 0B 00 19 77 3A : 93
0540 16 88 21 10 00 19 77 3A : 99
0548 17 88 21 11 00 19 77 3A : 9B
0550 18 88 21 08 00 19 77 3A : 93
0558 19 88 21 12 00 19 77 C9 : 2D
0560 CD 96 02 21 0C 00 09 5E : F9
0568 23 56 1B 72 2B 73 7B B2 : D1
0570 C2 8A 05 11 3C 00 73 23 : 34
0578 72 21 0E 00 09 5E 23 56 : 81
---------------------------------
SUM: 12 19 8D 07 CA F3 12 A9 E5CF

0580 E5 21 0A 00 19 EB E1 72 : 67
0588 2B 73 21 0E 00 09 5E 23 : 57
0590 56 ED 53 12 88 21 09 00 : 5A
0598 09 7E 32 10 88 21 0A 00 : 7C
05A0 09 7E 32 11 88 21 12 00 : 85
05A8 09 7E 32 14 88 C5 01 7C : 97
05B0 04 21 04 00 CD 3C 01 42 : 75
05B8 4B 11 0B 88 21 06 00 CD : E3
05C0 3C 01 C1 21 0B 00 09 7E : B1
05C8 C5 01 1A 88 21 04 00 CD : 5A
05D0 3C 01 C1 F6 F0 3C CA FB : E5
05D8 05 3D 16 04 21 36 06 0F : C8
05E0 D2 EC 05 23 23 15 C2 DF : BF
05E8 05 C3 FB 05 7E E5 21 10 : 5C
05F0 00 09 77 E1 23 7E 21 11 : 34
05F8 00 09 77 21 10 00 09 7E : 38
---------------------------------
SUM: E9 2E C3 AA 38 4C 4C F3 1E4D

0600 21 09 00 09 86 77 21 11 : 62
0608 00 09 7E 21 0A 00 09 86 : 41
0610 77 21 0E 00 CD 3C 01 AF : 5F
0618 32 04 88 ED 43 02 88 C5 : 3D
0620 01 05 88 21 0A 00 CD 3C : C2
0628 01 C1 3A 04 88 B7 C8 21 : 28
0630 00 00 CD 3C 01 C9 00 FF : D2
0638 00 01 FF 00 01 00 46 06 : 4D
0640 4C 06 52 06 58 06 0A 0D : 1F
0648 00 FF 40 30 1E 0C 00 01 : 9A
0650 2A 31 14 06 FF 00 24 32 : CA
0658 14 13 01 00 25 33 5E 06 : E4
0660 62 06 AB 01 72 06 C5 01 : 52
0668 92 06 76 06 5E 06 66 06 : E4
0670 02 00 11 1A 88 C9 F5 01 : 74
0678 00 1C 3E 07 ED 79 01 00 : C8
---------------------------------
SUM: 4C 6F B9 DC 13 C8 3B BB 9B40

0680 1B AF ED 79 F1 C6 0E 01 : F6
0688 00 1C ED 79 01 00 1B ED : 8B
0690 78 C9 C9 93 06 97 06 AB : EB
0698 01 A9 06 C5 01 41 07 D1 : 8F
06A0 06 AD 06 93 06 9B 06 02 : F5
06A8 00 11 1C 88 C9 C5 3A 1F : 9C
06B0 88 4F 6F 06 00 26 00 29 : 9B
06B8 29 09 29 29 29 3A 1E 88 : 8D
06C0 4F 06 00 09 01 00 30 09 : 98
06C8 4D 44 3A 20 88 ED 79 C1 : 9A
06D0 C9 C5 01 00 20 21 00 10 : E0
06D8 16 07 CD 0F 07 01 00 30 : 31
06E0 21 00 10 16 20 CD 0F 07 : 4A
06E8 3E 23 32 20 88 3E 00 32 : AB
06F0 1F 88 CD 19 07 3E 18 32 : 1C
06F8 1F 88 CD 19 07 3E 00 32 : 04
---------------------------------
SUM: 63 9C 47 34 57 F4 64 E3 3FE4

0700 1E 88 CD 2D 07 3E 27 32 : 3E
0708 1E 88 CD 2D 07 C1 C9 ED : 1E
0710 51 03 2B 7D B4 C2 0F 07 : 88
0718 C9 3E 27 32 1E 88 CD AD : 80
0720 06 3A 1E 88 A7 C8 3D 32 : C4
0728 1E 88 C3 1E 07 3E 18 32 : 16
0730 1F 88 CD AD 06 3A 1F 88 : 08
0738 A7 C8 3D 32 1F 88 C3 32 : 7A
0740 07 C9 00 00 00 00 00 00 : D0
0748 00 00 00 00 00 00 00 00 : 00
---------------------------------
SUM: 47 2C D7 8E B3 11 03 F1 AEF4
```

X68000 BASIC入門

最終回

必殺サンプリング戦法

Nakamori Akira
中森 章

約1年に渡ってご紹介してきたX68000BASIC入門も，ついに最終回を迎えました。今回は最後の砦ともいうべきADPCMに挑戦です。その昔，ダンシング・ヒーローを奏でていたX68000のビープ音に感動したという経験を生かして，中森氏はどのように料理するのでしょうか。

X68000が発売された当時，いちばん驚いたのは，ビープ音を自由に変更できる機能でした。ビープ音といえば，あの「ピーッ」という味気ない音が常識であると思い込んでいた私は，編集室のマシン室の X68000から流れてくる「ダンシング・ヒーロー」のイントロに大きなショックを受けたものです。そしてこのビープ音の変更はX68000の特徴のひとつであるADPCM（Adaptive Differential Pulse Code Modulation）方式を用いたサンプリング機能によって行われていたのでした。

おそらく，X68000はサンプリング機能を備えた世界で最初のパソコンです。サンプリング機能自体は新しい技術ではありませんが，それとパソコンとの合体はまさに新機軸だったといえるでしょう。このサンプリング機能によってパソコンの可能性が大きく広がったということは疑う余地はありません。とにかく，「ダンシング・ヒーロー」のイントロとの出会い。これがなければ私はX68000を買っていなかったかもしれません(そんなことないってば)。

X-BASICの基礎事項（前回まで）

X-BASIC では変数を使用する前には変数の型宣言をしなければなりません。宣言できるデータ型はint（4バイト整数），char（1バイト整数），str（文字列），float（実数）の4種類です。

X-BASIC のプログラムの実行はその大部分が関数の呼び出しによって行われます。それ以外は制御構造です。型宣言と制御構造と関数，これがX-BASICの3大要素です。

X-BASIC には画面上のキャラクタをスムーズに移動させるためのスプライト機能が備わっています。これにより最大128個のキャラクタを同時に移動させることができます。この移動のときパターンの反転，色の変更なども可能です。また，バックグラウンドと呼ばれる画面が2面あり，ここでは最大64×64個並べたキャラクタを背景として利用できます。バックグラウンド面上では，画面上のすべてのキャラクタが同時に移動します。

また，X-BASICでは65536色同時発色を特徴とするX68000のグラフィック機能を扱うことができます。色数が65536色であるのはグラフィック画面（実画面）が512×512ドットの場合ですが，色数を256色，16色と減らすことによって，実画面を2画面，4画面と増やすことができます。さらに，色数を16色，実画面数を1画面に限れば1024×1024ドットという大画面を扱うこともできます。また，複数個の実画面は高速に切り換えることができますし，それぞれをスクロールさせることもできます。この機能をうまく使えば，アニメーションも簡単です。

また，グラフィック画面の特徴として半透明機能があります。これは，グラフィックの実画面同士あるいはグラフィック画面とテキスト画面（スプライト画面）を重ね合わせて表示する機能です。この重ね合わせは，最も優先順位の高いグラフィック画面が半透明になることで実現されます。しかし，残念ながら半透明機能はX-BASIC から直接扱うことができません。メモリ上にマッピングされている，X68000のビデオコントローラの内部レジスタを直接書き換えることで扱うことができます。

X68000ではグラフィック画面のみならず，テキスト画面もビットマップ方式を採用しています。さらに，テキスト画面は16色のパレットやスクロール機能も備わっています。このため，テキスト画面もグラフィック画面と対等に扱うことができます。たとえば，グラフィック画面の退避領域としてテキスト画面を使用することができます。

また，X68000にはマウスが標準で付いてきます。そして，X-BASIC ではこのマウスを扱うための関数が用意されています。マウスを入力装置とすることで操作性のよいプログラムを書くことができます。

X68000のハードウェアでスプライト，グラフィックと並ぶ3大特徴のひとつがFM音源です。X68000はFM音源用のLSIとしてOPM(YM2151)を内蔵し，8オクターブ，8重和音のステレオ演奏を行うことができます。そして，X-BASIC はOPMに音楽を演奏させるためのインタフェイスとしてMML（ミュージック・マクロ・ランゲージ）と呼ばれる言語が用意されています。このMMLは演奏の繰り返し指定を簡単に記述できるという特徴があります。また，この音楽の演奏は，割り込みによって，ほかのプログラムの実行と同時に行うことが可能なため，FM音源をゲームなどのBGMとして利用することもできるでしょう。

サンプリングとADPCM

サンプリング機能とは，その名のとおり，実際の音をサンプリング（標本を取るという意味，早い話が録音）して記憶しておき，それをそのまま再生することにより元の音を再現する機能です。ADPCM方式というのはその録音の方式です。X68000には音を出す機能としてFM音源もありますが，FM音源がサイン波を組み合わせて音を人工的に作り出すのに対して，サンプリングは実際に鳴っている音をそのまま記憶してオウム返しに出力する点で異なっています。

たとえば，FM音源を用いて本物の楽器に近い音を作るためにはかなりの労力が必要ですが，サンプリングを用いればなんの努力も必要なく本物の楽器の音が作り出せてしまうのです（録音してるのだから当たり前か）。

それでは，これからサンプリング機能とADPCMについての基礎知識を説明していきましょう。まずはサンプリング周波数です。音声のデータ（音の波形だと思ってください）は，時間が経過するにつれて連続的（アナログ的）に変化していますが，それをパソコンのメモリ内に取り込むには，音声のデータをデジタルな0と1の並びに変換する必要があります。その方式のひとつがPCM方式と呼ばれるものなのです。

PCM（Pulse Code Modulation）とは，ごく短い一定間隔おきに音声のデータをサンプリングし，そのとき得られた値を0と1の並びに変換(A/D変換)して記憶するという方式です。PCM方式による録音は，アナログ形式に比べて，S/N比の向上，ダイナミックレンジの拡大，録音時の音質劣下防止という点に特徴があるといわれています。そして，PCM方式のサンプリングの間隔を示す指数がサンプリング周波数と呼ば

れるものです。

サンプリング周波数は1秒間にサンプリングされるデータの個数を示すもの（つまり何回サンプリングするか）で，たとえば，サンプリング周波数が15.6KHzであれば1秒間に15600個のデータが等間隔の時間で取り込まれることを示します(図1)。X68000ではADPCM録音のサンプリング周波数として，

3.9KHz　5.2KHz　7.8KHz
10.4KHz 15.6KHz

という5種類の周波数をサポートしています。サンプリングは音声のデータを0と1の並びに変換しますが，そこにはサンプリング周波数の情報は残りません。このため，録音されたデータを正しく再生するためには，録音時と再生時に同一のサンプリング周波数を指定する必要があります。しかし，このことを逆に利用して，15.6KHzのサンプリング周波数で録音したデータを7.8KHzなどの周波数で再生して音程を変えるということもできます（本来の1/2のサンプリング周波数で再生すると音程は1オクターブ下がる)。

さて，サンプリングする音を本当の音に近付けるためには，サンプリング周波数を高くする（1秒間にサンプリングするデータ数を増やす）に越したことはありません。しかし，音声のデータをそのまま記憶していたのではメモリやディスクの容量があっという間にパンクしてしまいます。そこで考えられたのが，前回サンプリングした値との差分を記憶していくDPCM (Differential PCM) です。

音声が通常は連続的に変化することを考えると，ある時点でサンプリングした音声データの値は前回サンプリングした値と似ていると予想されます。したがって差分は0に近い値になり，それを記憶するためのメモリやディスクの容量も小さくてすんでしまうのです。これがADPCMのDPCMです。

そして，ADPCMのAというのは「適応制御（Adaptive Control)」からくるものです。これはDPCMによって得られる差分のデータを適当に変換して，もっと少ないデータで間に合うようにする技法です。X68000のADPCMでは具体的にどのような適応制御が行われているのかは知りませんが，X68000では1回のサンプリングで得られるデータを4ビット（1ビットは符号ビット）という少ないビット数で表現できるようになっています(図2)。それでいて，なお再生時に本物に近い音が出るのはADPCMのAのおかげなのです。マニュアルにADPCMのサンプリング周波数と1秒間に消費されるメモリ容量の記述がありますが，1回のサンプリングで1/2バイト(4ビット）のデータが消費されることを知っていれば，サンプリング周波数の値を2で割ったものがそれになることがわかりますね。

もう少し詳しく知りたい人は特集の加藤氏の記事を参照するとよいでしょう。

AUDIO.FNCの中身

X-BASIC でサンプリング関係の関数はAUDIO.FNCのなかに入っています。もうほとんどワンパターンになってしまいましたが，例によって，これを以前この連載で紹介した（1987年10月号）プログラムでダンプしてみましょう。AUDIO.FNCのなかに定義されている関数は次の6つです。

a_play(*ptr, char, char, [int])
a_rec(*ptr, char, [int])
a_stat( )
a_end( )
a_stop( )
a_cont( )

注　( ) 内は引数の型
　　[ ] 内は省略可能な引数の型
　　*ptrは1次元配列へのポインタ

ここで，「おやっ」と思った人がいるかもしれませんね。マニュアルにはa_playとa_rec以外の関数は載っていません。それどころか，バージョン2よりも前のX-BASICに付属のAUDIO.FNCにはa_playとa_recの2つの関数しか定義されていないのです。なにか初期のX68000を買った人は損をした気がしないでもありませんが，マニュアルに載ってないくらいの関数ですから使い道はそれほどありません。しかし，これらの関数を外部関数として定義するのは簡単なので，あとで作ってみます。とりあえず，これらの関数の機能を書いておくと次のようになります。

a_play　ADPCMデータの再生
a_rec　ADPCMデータの録音
a_stat　ADPCMの動作状況
a_end　ADPCMの強制終了
a_stop　ADPCMの動作中断
a_cont　ADPCMの動作再開

これを見てわかるように，バージョン2のX-BASICで追加された関数は，FM音源の制御であったのと同様なADPCMの中断，再開，終了に関するものです。しかし，a_play関数の実行時（ADPCMの再生時）は，通常の場合は，演奏が終わるまでBASICに制御が戻ってきません。このため，ADPCMの中断や再開をする関数があってもあまり嬉しくはないのです（演奏中は実行できない)。a_play関数の実行後，すぐBASICに

図1　サンプリングとサンプリング周波数

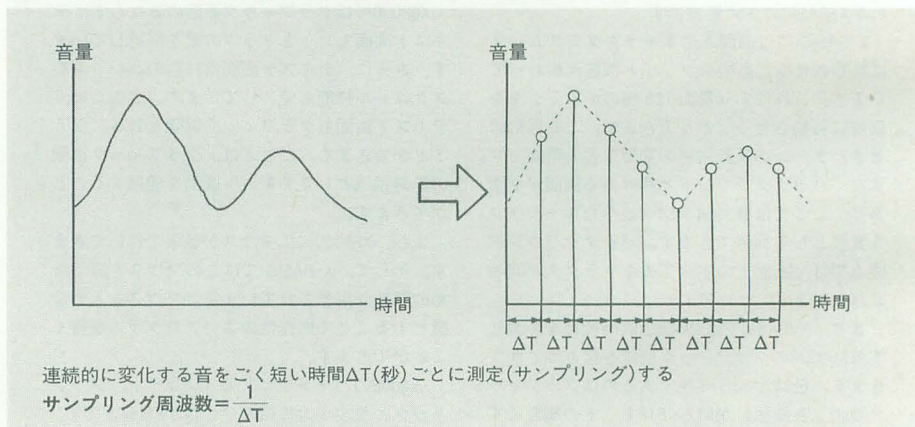
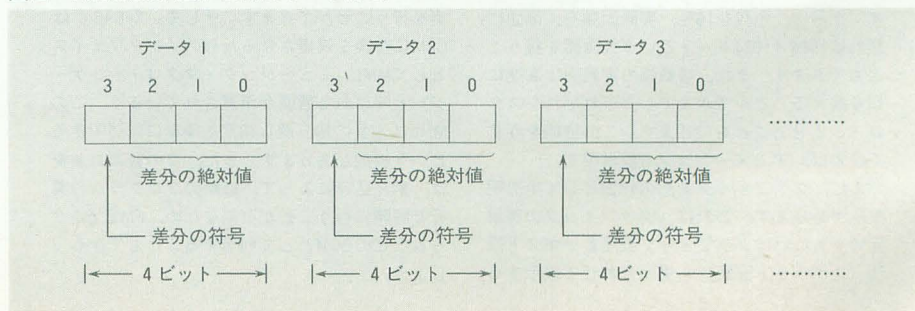



図2　X68000のADPCMのデータ形式

制御が戻ってくる場合はADPCMのデータが0FF00Hバイト以下のとき（DMA 転送の都合）ですが，経験的にはa_stop関数とa_cont関数はビープ音に対してしかうまく働かないようです。結局，あまり使えない関数であることがわかるでしょう。

使い道がないといった手前恐縮ですが，それでは，待望（？）の外部関数です。a_stat，a_stop，a_cont，a_end を外部関数で定義するとリスト1のようになります。リスト1を見てもらえばわかりますが，これらの関数はすべてIOCS コールによって実現されています。プログラムの構造はあまりにも単純なので説明は不要でしょう。なおリスト1の関数名はバージョン2のX-BASICのAUDIO.FNCと共存できるように

a_stat1　a_stop1　a_cont1　a_end1

となっています。外部関数の X-BASIC への取り込み方法は，この連載で何回もやっていますから，バックナンバーを参照してください。

ADPCMを使う

ADPCMの利用で必要な関数はa_play関数とa_rec関数です。すなわち，音を録音（a_rec）して再生（a_play）すること，それがすべてです。ADPCMによる録音と再生は次の手順で行われます。

1）PCMデータを格納するための配列を宣言する
2）a_rec関数を実行する（録音）。録音する音声の入力はX68000のAUDIO IN 端子にラジカセなどの出力端子を接続して行う
3）a_play関数を実行する（再生）。音声の出力はX68000の内蔵スピーカー，またはAUDIO OUT端子から行う
4）必要な場合においてPCMデータをファイルに格納する

また，ファイルにあらかじめ格納されているPCMデータを再生するためには次の手順になります。

1）PCMデータを格納するための配列を宣言する
2）ファイルの内容を配列に読み込む
3）a_play関数を実行する（再生）

そして，録音と再生時に使用する a_rec 関数とa_play関数のフォーマットは以下のようになっています。

a_rec(na, sf[, lng])

na　PCMデータを格納する配列名
sf　サンプリング周波数
　0　3.9KHz
　1　5.2KHz
　2　7.8KHz
　3　10.4KHz
　4　15.6KHz
lng　録音する配列naの長さ（省略可能）
　0～配列naの添字の最大値＋1
　省略時はnaの全データ

a_play(na, sf, md[, lng])

na　PCMデータを格納している配列名
sf　サンプリング周波数
　0　3.9KHz
　1　5.2KHz
　2　7.8KHz
　3　10.4KHz
　4　15.6KHz
md　出力モード
　0　出力なし
　1　左のみ出力
　2　右のみ出力
　3　左右から出力
lng　再生する配列naの長さ（省略可能）
　0～配列naの添字の最大値＋1
　省略時はnaの全データ

関数の使い方を頭に叩き込んだところで実際にプログラムを作ってみましょう。といっても，単にa_rec関数を実行して a_play関数を実行するだけではあまりにも能がないので少し工夫をします。

ADPCMで録音するためにはX68000のAUDIO IN端子から音声を入力しますが，a_rec関数を実行するまではいまどんな音がAUDIO IN端子から入力されているか知ることができません（a_rec関数の録音中には内蔵スピーカーなどから音が出る）。これでは歌謡曲などの音楽を途中から録音したいときは，どこからa_rec 関数を実行したらよいかわからず困ってしまいます。そこで，a_rec関数を実行する前でもAUDIO IN 端子に入力されている音声を聞くことができるようにします。これはなにか高級なことをやらなければならないように思うかもしれませんが，非常に単純なことです。タネを明かせば「非常に短いデータ長を使ってa_recを繰り返す」ということをやればいいのです。つまり，以下のようなアルゴリズムです。

1）a_rec関数で少しだけ録音する
2）キー入力を調べる
　キー入力があれば4）へ
3）1）へ戻って再び繰り返す
4）a_rec関数で実際に録音する
5）a_play関数で再生する

本当にやりたいのは4）と5）の操作なのですが，それに先立って1）～3）のキー入力待ちのループがあります。この1）～3）のループ内で a_rec 関数を少しだけ実行するのです。つまり，a_rec関数に与えるPCMデータ用の配列の大きさを小さくするわけです。

たとえば配列の大きさが500バイトでサンプリング周波数が15.6KHz（1秒間に7800バイトを消費）の場合，a_rec関数の実行に要する時間は，

リスト1　4つの外部関数定義プログラム

```
        .nlist
void_ret        equ     $ffff
int_ret         equ     $8001
_ADPCMSNS       equ     $66
_ADPCMMOD       equ     $67
        .list
********************************
* インフォメーション　テーブル *
********************************
        dc.l    _ret,_ret,_ret,_ret
        dc.l    _ret,_ret,_ret,_ret
        dc.l    _token,_param,_exec
        dc.l    0,0,0,0,0
**************
* プログラム *
**************

****************************************
*       何もしない                     *
****************************************
        .even
_ret:
        rts
****************************************
*       ＡＤＰＣＭの動作中断           *
****************************************
        .even
_a_stop:
        moveq.l #1,d1
        moveq.l #_ADPCMMOD,d0
        trap    #15
        rts
****************************************
*       ＡＤＰＣＭの動作状況           *
****************************************
        .even
_a_stat:
        moveq.l #_ADPCMSNS,d0
        trap    #15
        move.l  d0,_ret_val
        clr.l   d0
        lea     _ret_arg,a0
        rts
****************************************
*       ＡＤＰＣＭの動作再開           *
****************************************
        .even
_a_cont:
        moveq.l #2,d1
        moveq.l #_ADPCMMOD,d0
        trap    #15
        rts
****************************************
*       ＡＤＰＣＭの強制終了           *
****************************************
        .even
_a_end:
        clr.l   d1
        moveq.l #_ADPCMMOD,d0
        trap    #15
        rts

****************
* データエリア *
****************
        .even
_token:
        dc.b    'a_stop1',0
        dc.b    'a_stat1',0
        dc.b    'a_cont1',0
        dc.b    'a_end1',0
        .even
_param:
        dc.l    _stop_param,_stat_param
        dc.l    _cont_param,_end_param
        .even
_stop_param:
_cont_param:
_end_param:
        dc.w    void_ret
_stat_param:
        dc.w    int_ret
        .even
_exec:
        dc.l    _a_stop,_a_stat
        dc.l    _a_cont,_a_end
        .even
_ret_arg:
        dc.w    0,0,0
_ret_val:
        dc.l    0

        end
```

500/7800＝0.06(秒)

です。したがって1)～3)のループでは0.06秒間音が鳴っては(a_rec関数の実行)，音が一瞬とぎれる(a_rec関数の終了)という動作を繰り返すのです。この間音は連続して鳴っているように聞こえます。このときキー入力があるとループを抜け出し，実際の録音が開始されます。タイミングによってはキーを押した瞬間の0.06秒分のデータを取り逃す恐れもありますが，まあ，それくらいは我慢しましょう。以上のような考えで作ったプログラムがリスト2です。

リスト2のプログラムではサンプリング周波数を15.6KHzに固定していますが，いろいろなサンプリング周波数を試したい人は適当にプログラムを変更してください。また，リスト2のプログラムで配列aの内容をファイルにセーブすればビープ音用のPCMデータを作ることができます(この場合のサンプリング周波数は15.6KHzから変えないでください)。

音声合成に挑戦

ADPCMとしてはa_rec関数で録音されたPCMデータをa_play関数で再生するのがノーマルな使用法です。しかし，ここではa_play関数で再生するPCMデータを人工的に作り出して，一種の音声合成に挑戦してみることにします。X68000のPCMデータは4ビットの差分データが並んだものです。これと同じデータを作り出せばよいわけです。といっても，人間の声などのように複雑な波形を持つ音のデータを作るのはやさしいことではありません。そこで，ひとつのサイン波によって作られる音のデータを作ります。これならばある時刻を与えたときの音の大きさ（これがサンプリングされる値）が簡単に計算できますからPCMデータを容易に作ることができます。

それでは，

$$F=A\cdot\sin(\omega\cdot t)$$

A　振幅
ω　振動数
t　時間

という波形を考えてみましょう。これはωという振動数で鳴り響く音になります。音階の「ド・レ・ミ……」は，それぞれ一定の振動数を持った音のことですから，いろいろな振動数を持つサイン波で，PCMデータを作ってやれば，ADPCMで「ド・レ・ミ……」を演奏することも不可能ではありません。たとえば，振動数ωを持つ音を鳴らすためのPCMデータは以下の3段階の手順で作り出すことができます。

1) サンプリングされた音の大きさA_0, A_1, A_2, A_3, ……を計算する

$$A_0=A\cdot\sin(2\pi\cdot\omega\cdot 0\cdot\Delta T)$$
$$A_1=A\cdot\sin(2\pi\cdot\omega\cdot 1\cdot\Delta T)$$
$$A_2=A\cdot\sin(2\pi\cdot\omega\cdot 2\cdot\Delta T)$$
$$A_3=A\cdot\sin(2\pi\cdot\omega\cdot 3\cdot\Delta T)$$
$$\vdots$$

ただし，ΔT：サンプリングの間隔（サンプリング周波数の逆数）

2) 音の大きさA_0，A_1，A_2，A_3，……からその差分D_0，D_1，D_2，D_3，……を計

リスト2　ADPCMによる録音と再生

```
 10 /*
 20 /*          ＡＤＰＣＭによる録音と再生
 30 /*
 40 dim char  a(65535)
 50 dim char  a0(499)    :/* ダミーの配列
 60 str ans
 70 /*
 80 /* キー入力があるまで a0 で録音・再生を繰り返す
 90 /*
100 a_rec(a0,4)
110 color 2
120 print "［Ｙ］キーを押すと録音を開始します。"
130 color 3
140 print
150 while 1
160   ans=inkey$(0)
170   if (ans="y")or(ans="Y") then break
180   a_rec(a0,4)
190 endwhile
200 /*
210 /* 本当に録音を開始する
220 /*
230 print "ＡＤＰＣＭの録音中です。" : print
240 a_rec(a,4)
250 print "録音を終了しました。"      : print
260 /*
270 /* a0 と a の配列の内容をつなげる
280 /*
290 /*
300 /* ＰＣＭデータを再生する
310 /*
320 color 2
330 print "［Ｙ］キーを押すと再生を開始します。"
340 color 3 : print
350 while 1
360   ans=inkey$(0)
370   if (ans="y")or(ans="Y") then break
380 endwhile
390 print "ＡＤＰＣＭの再生中です。": print
400 a_play(a,4,3)
410 print "再生を終了しました。"
420 end
```

リスト3　ADPCMデータ作成プログラム

```
 10 /*
 20 /*          ＡＤＰＣＭデータ作成プログラム
 30 /*
 40 float A=440#
 50 float freq
 60 float dt=6.41025641025e-5#          /* 1/15.6K
 70 float a0,a1
 80 int   d0,d1
 90 dim   char   N(8000)
100 str   fil,num
110 int   fp
120 int   i,j
130 /*
140 for i=0 to 17
150   freq=A*exp(i*log(2#)/12#)
151   locate 0,0 : print chr$(5)+"freq=";freq
160   a0=0#
170   for j=1 to 8000
171       locate 0,1 : print chr$(5);j
180       a1=7#*sin(pi(2)*dt*j*freq)
190       d1=int(a1-a0)
200       if((j mod 2)=0) then {
210             N((j¥2)-1)=(encode(d1) shl 4) or (encode(d0))
220       }
230       d0=d1 :  a0=a1
240    next
250    num=str$(i) : if i<10 then num="0"+num
260    fil="D:A"+num+".DAT"
270    fp=fopen(fil,"c") : fwrite(N,8001,fp) : fclose(fp)
280 next
290 end
300 /*
310 func int encode(x)
320  int s=8
330  if(x>0) then s=0
340  return(s or (x and  7))
350 endfunc
```

算する。

$D_0 = A_1 - A_0$

$D_1 = A_2 - A_1$

$D_2 = A_3 - A_2$

$D_3 = A_4 - A_3$

⋮

3) 差分D_0, D_1, D_2, D_3, ……を4ビットで表しPCMデータに変換する。このデータを配列やファイルに格納するときは2個(8ビット)ずつ組にしてバイト単位で行う。データ形式は図2を参照のこと

ところで，A(ラ)の音は440Hzの振動数を持つといわれています。そこで，先の1)～3)でωを440にすれば，再生するとAの音で鳴るPCMデータを作ることができるのです。Aの音の振動数を知っていれば，ほかの音の振動数は12平均律（1オクターブ内に等しい振動数の間隔で12個の半音を配置する調律法）によって

A	440
A♯	$440\times(\sqrt[12]{2})^{1}$
B	$440\times(\sqrt[12]{2})^{2}$
C	$440\times(\sqrt[12]{2})^{3}$
C♯	$440\times(\sqrt[12]{2})^{4}$
D	$440\times(\sqrt[12]{2})^{5}$
D♯	$440\times(\sqrt[12]{2})^{6}$
E	$440\times(\sqrt[12]{2})^{7}$
F	$440\times(\sqrt[12]{2})^{8}$
F♯	$440\times(\sqrt[12]{2})^{9}$
G	$440\times(\sqrt[12]{2})^{10}$
G♯	$440\times(\sqrt[12]{2})^{11}$

A（1オクターブ上）

$$440\times(\sqrt[12]{2})^{12}=440\times2$$

という計算式で求めることができます。このとき2の12乗根のベキ乗を求めることが必要になりますが，これは指数と対数の，

$$x=\exp(\log(x))$$

という関係を用いれば，

$$(\sqrt[12]{2})^{n}=\exp((n/12)*\log(2))$$

という式から求めることができます。以上のような考えで，各音階に対応したPCMデータを作るためのプログラムがリスト3です。リスト3を実行するとドライブCに

A＊＊. DAT（＊＊は2桁の数字）

というファイルが作られていきます。ドライブやファイル名が気に入らないときは行番号260で作っているファイル名(filという変数)を変更してください。作られるファイルと音階の対応は，

A00. DAT→A	A01. DAT→A♯
A02. DAT→B	A03. DAT→C
A04. DAT→C♯	A05. DAT→D
A06. DAT→D♯	A07. DAT→E
A08. DAT→F	A09. DAT→F♯
A10.DAT→G	A11.DAT→G♯
A12.DAT→A	A13.DAT→A♯
A14.DAT→B	A15.DAT→C
A16.DAT→C♯	A17.DAT→D

となっています。

リスト4 ADPCMによる「はとぽっぽ」

```
 10 /*
 20 /*      「はとぽっぽ」 by ADPCM
 30 /*
 40 /*
 50 /* A00:A  A01:A+  A02:B  A03:C  A04:C+  A05:D  A06:D+  A07:E
 60 /* A08:F  A09:F+  A10:G  A11:G+ A12:A   A13:A+ A14:B   A15:C....
 70 /*
 80 dim char A00(8000),A01(8000),A02(8000),A03(8000),A04(8000),A05(8000)
 90 dim char A06(8000),A07(8000),A08(8000),A09(8000),A10(8000),A11(8000)
100 dim char A12(8000),A13(8000),A14(8000),A15(8000),A16(8000),A17(8000)
110 /*
120 fp=fopen("B:A00.DAT","r") : fread(A00,8001,fp) : fclose(fp)
130 fp=fopen("B:A01.DAT","r") : fread(A01,8001,fp) : fclose(fp)
140 fp=fopen("B:A02.DAT","r") : fread(A02,8001,fp) : fclose(fp)
150 fp=fopen("B:A03.DAT","r") : fread(A03,8001,fp) : fclose(fp)
160 fp=fopen("B:A04.DAT","r") : fread(A04,8001,fp) : fclose(fp)
170 fp=fopen("B:A05.DAT","r") : fread(A05,8001,fp) : fclose(fp)
180 fp=fopen("B:A06.DAT","r") : fread(A06,8001,fp) : fclose(fp)
190 fp=fopen("B:A07.DAT","r") : fread(A07,8001,fp) : fclose(fp)
200 fp=fopen("B:A08.DAT","r") : fread(A08,8001,fp) : fclose(fp)
210 fp=fopen("B:A09.DAT","r") : fread(A09,8001,fp) : fclose(fp)
220 fp=fopen("B:A10.DAT","r") : fread(A10,8001,fp) : fclose(fp)
230 fp=fopen("B:A11.DAT","r") : fread(A11,8001,fp) : fclose(fp)
240 fp=fopen("B:A12.DAT","r") : fread(A12,8001,fp) : fclose(fp)
250 fp=fopen("B:A13.DAT","r") : fread(A13,8001,fp) : fclose(fp)
260 fp=fopen("B:A14.DAT","r") : fread(A14,8001,fp) : fclose(fp)
270 fp=fopen("B:A15.DAT","r") : fread(A15,8001,fp) : fclose(fp)
280 fp=fopen("B:A16.DAT","r") : fread(A16,8001,fp) : fclose(fp)
290 fp=fopen("B:A17.DAT","r") : fread(A17,8001,fp) : fclose(fp)
300 /*
310 /* 演奏
320 /*
321 for i=1 to 2
330         a_play(A03,4,3,4000)
340         a_play(A05,4,3,4000)
350         a_play(A07,4,3,8000)
360         a_play(A10,4,3,2000)
370         a_play(A07,4,3,2000)
380         a_play(A03,4,3,4000)
390         a_play(A05,4,3,8000)
400         a_play(A03,4,3,4000)
410         a_play(A05,4,3,2000)
420         a_play(A07,4,3,2000)
430         a_play(A10,4,3,2000)
440         a_play(A10,4,3,2000)
450         a_play(A07,4,3,2000)
460         a_play(A07,4,3,2000)
470         a_play(A10,4,3,2000)
480         a_play(A07,4,3,2000)
490         a_play(A03,4,3,2000)
500         a_play(A05,4,3,2000)
510         a_play(A07,4,3,8000)
520         a_play(A10,4,3,2000)
530         a_play(A10,4,3,2000)
540         a_play(A10,4,3,2000)
550         a_play(A07,4,3,2000)
560         a_play(A12,4,3,2000)
570         a_play(A12,4,3,2000)
580         a_play(A12,4,3,2000)
590         a_play(A10,4,3,2000)
600         a_play(A07,4,3,2000)
610         a_play(A07,4,3,2000)
620         a_play(A07,4,3,2000)
630         a_play(A05,4,3,2000)
640         a_play(A03,4,3,8000)
650        for j=0 to 8000 :next
660 next
```

さて音階のPCMデータができたらそれを使って音楽を演奏してみなければ面白くありません。演奏をさせるためのサンプルプログラムがリスト4です。ここで演奏する曲目は「はとぽっぽ」です。リスト4のプログラムでやっていることは，リスト3のプログラムで作ったPCMデータをファイル(ドライブBからになっているので注意)から配列に読み込んで，a_play関数で演奏しているだけです。このとき，a_play関数で再生する長さを変える(第4引数)ことによって，演奏する音の長さを変えています。

つまり，2分音符を8000として，4分音符4000，8分音符2000，16分音符1000という具合いに分割しています。ここでは例として「はとぽっぽ」を取り上げましたが，ほかの曲ももちろん演奏することができるでしょう。ADPCM版MMLの出来上がりというわけです。まあ，実際にリスト4のプログラムを実行してみてください。

ちょっと単調だけど結構，聴ける演奏になっていますね。最初，ADPCMのAがなんだかよくわからなかったので，DPCMだけでPCMデータを作ってみましたが，こんなにうまくいくとは意外でした。

FM音源をサンプリング

リスト3では，人工的にPCMデータを作り出したわけですが，やはりADPCMの本道(?)は，実際にサンプリングしたPC

Mデータを用いることでしょう。

それでは，リスト4でファイル（A00.DAT～A17.DAT）から読み込むPCMデータをFM音源で作ることにします。これなら，まともな演奏が期待できそうですね。ここではFM音源で演奏した「ド・レ・ミ……」の音をテープレコーダに録音し，その音をa_rec 関数で取り込むということをやってみます。FM 音源で音を鳴らすためのプログラムがリスト5です。

このプログラムを実行するとAの音から1オクターブ上のDの音までが順番に演奏されます。ただそれをテープレコーダで録音してください。この音はプリセット音の「パイプオルガン」で，コーラス効果を出すために左右から出力される音の振動数を少しずらしてあります。テープレコーダへの録音が終わったらそれを再生させ，リスト6のプログラムで PCM データに変換します。

リスト6のプログラムは基本的にはリスト2のプログラムと一緒で，X68000のAUDIO IN端子に入力される音を短いa_recの実行をループしながら鳴らしておき，キーが押されると本当のa_rec 関数の実行を行うというものです。そして，この動作を18回（作るPCMデータのファイル数）繰り返すようになっています。私たちがやることは，リスト6のプログラムを実行したあと，録音していたテープを再生し，テープから流れてくる音の音程が変わったらYキーを押すという操作を18回行うだけです。18回キーを押し終わったあとにはドライブCに18個のPCMデータのファイルができているでしょう（ねっ，簡単でしょう)。

ファイルの作られるドライブやファイル名を変更したい人は，例によって行番号190を書き換えてください。リスト6のプログラムによって作られる PCM のデータはリスト4のプログラムでそのまま使うことができますから，このデータを使ってリスト4の「はとぽっぽ」を再演奏してみましょう。今度はまともな「はとぽっぽ」を聴くことができると思います。また，リスト6のプログラムで「あ・い・う……」といった人の声のPCMデータを作って，X68000におしゃべりさせることもできますね。

リスト5　FM音源で音を鳴らす

```
 10 /*
 20 /*  Ｆ Ｍ 音 源 で 音 を 鳴 ら す （ だ け ）
 30 /*
 40 m_init()
 50 m_alloc(1,1000) : m_alloc(2,1000)
 60 m_assign(1,1)   : m_assign(2,2)
 70 m_trk(1,"L1V12P1@15 Y48,0  >A&ARA+&A+RB&BR<C&CRC+&C+RD&DRD+&D+RE&ER")
 80 m_trk(2,"L1V12P2@15 Y49,40 >A&ARA+&A+RB&BR<C&CRC+&C+RD&DRD+&D+RE&ER")
 90 m_trk(1,"L1V12@15 F&FRF+&F+RG&GRG+&G+RA&ARA+&A+RB&BR<C&CRC+&C+RD&DR")
100 m_trk(2,"L1V12@15 F&FRF+&F+RG&GRG+&G+RA&ARA+&A+RB&BR<C&CRC+&C+RD&DR")
110 m_play(1,2)
```

リスト6　ADPCMデータ作成プログラム

```
 10 /*
 20 /*          ＡＤＰＣＭデータ作成プログラム
 30 /*
 40 dim   char   N(8000)
 50 str   fil,num
 60 int   fp
 70 int   i
 80 /*
 90 for i=0 to 17
100   print i,"番目：［Ｙ］キーを押すと録音を開始します。"
110   a_rec(N,4,200)
120   while 1
130         fil=inkey$(0)
140         if (fil="y")or(fil="Y") then break
150         a_rec(N,4,200)
160   endwhile
170   a_rec(N,4,8001)
180   num=str$(i) : if i<10 then num="0"+num
190   fil="C:B"+num+".DAT"
200   fp=fopen(fil,"c") : fwrite(N,8001,fp) : fclose(fp)
210   print "完了しました。"
220 next
230 end
```

ビープ音変更プログラム

私がADPCMに感動したのはビープ音によってでした。そこで，ビープ音を扱うプログラムを作らないのは片手落ちというものです。ここではビープ音を変更するプログラムを作ってみましょう。ビープ音はデバイスドライバ（BEEP.SYS）によってX68000の起動時に設定されます。このため，ビープ音を一度設定したらそう簡単に変更することはできません。このため，X-BASICのBEEP命令を実行すると，いつも同じビープ音がなるのです(当たり前のことですが)。これでは面白くないので，ビープ音を何度でも設定し直せるプログラムを作ります。といっても，これはX-BASICだけでは不可能です。そこで，外部関数の助けを借りることにします。

それでなにをすればいいかというと，ことは簡単です。ビープ音に関しては，ビープ音のためのPCMデータが格納されているメモリの先頭番地（32ビット）が978H番地に，PCMデータのサイズ（バイト数，16ビット）が97CH番地に格納されています。この978H，97CHという番地はBIOS ROM の内部で絶対アドレスで指定してありますから，ROMのバージョンが変更にならない限り変わることはないので安心です。プログラムでは，これらのアドレスの内容を書き換えてやればいいのです。このため，外部関数の仕様はPCMデータの格納された配列の名前と，配列のサイズを与えることになります。つまり，ビープ音を変更する外部関数の名前をchg_beepとして，そのフォーマットを次のように決めます。

chg_beep(na[, lng])

na	ビープ音用の PCM データを格納している配列名
lng	ビープ音を鳴らす配列naの長さ（省略可能)。バイト数で指定すること。省略時はnaの配列の全データ

ここでいう配列とは1次元配列（charかintかfloat)のことで，配列名を引数とする場合は，外部関数内では配列をポインタ（アドレス）の形式で受け取ることになります。そこで1次元配列のポインタがどのようなものであるかを知っておく必要があります。手元にX-BASICの内部資料があるわけではありませんから，本当のところは不明ですが，配列へのポインタは図3のような情

不思議だけどできるのです

a_rec関数の実行が終わるとAUDIO IN端子から入力されている音声は聞こえなくなってしまいます。しかし，a_rec 関数の実行後もAUDIO IN端子から入力されている音声を鳴らし続ける方法があります。それはa_rec関数で録音する PCM データの格納される配列の大きさを0FF00Hバイトにすることです。つまり，宣言としては

dim char a(&HFEFF)

という具合になります。0FF00Hバイトというのは DMA転送の処理単位ですが，この境界値のところになにかがあるのかもしれませんね。

報を示していると思われます。

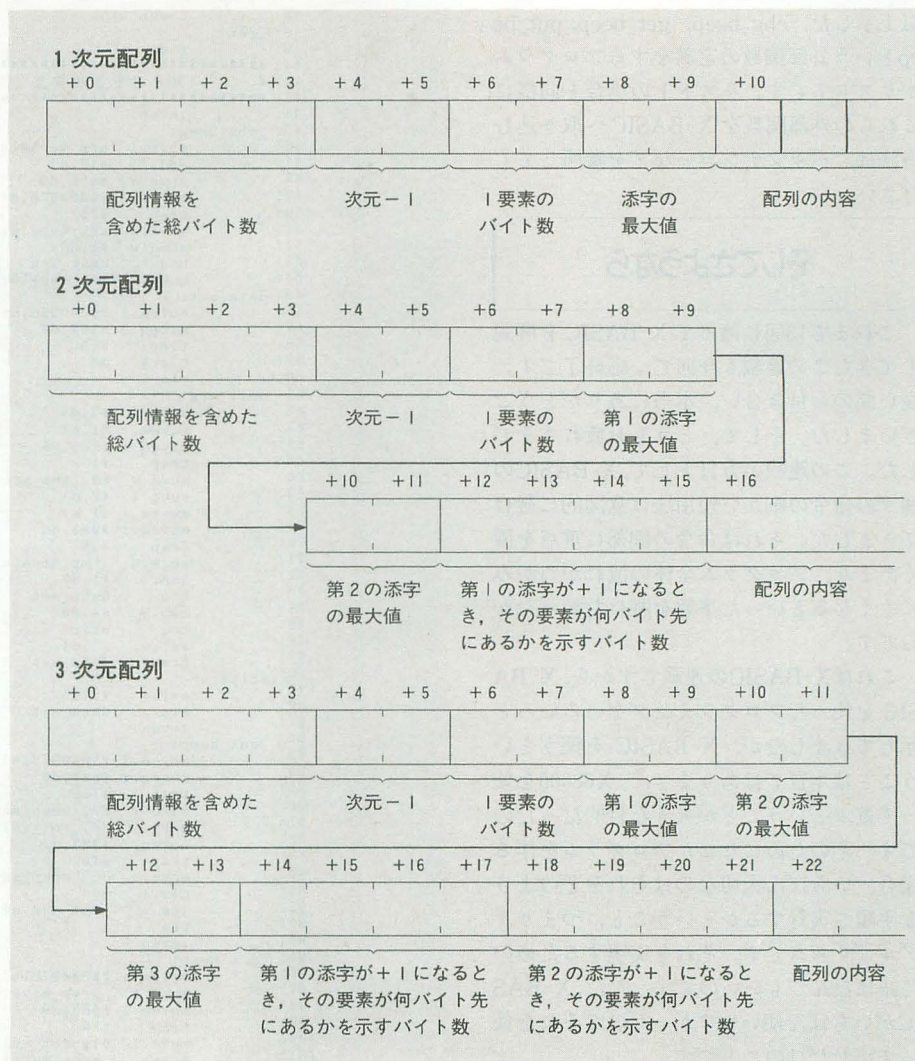
これは配列の次元が変わると内容も微妙に変わっています。つまり，1次元配列の場合は，10バイトの配列情報に続いて実際の要素がメモリ内に格納されており，その10バイトの配列情報の先頭アドレスを示すものが配列へのポインタなのです。したがって，配列の要素が格納されている先頭番地を知るためにはポインタに10を加える必要があります。これだけのことを知っていれば外部関数が書けそうです。外部関数でやるべきことは次のようになります。

1) 引数として配列へのポインタとサイズを受け取る
2) 配列へのポインタに10を加えたものを978H番地に格納する
3) サイズを97CH番地に格納する。サイズが省略されたときは，配列へのポインタに8を加えたアドレスから要素数（正確には配列の添え字の上限）を得て，それに1要素のサイズを掛け算した値を97CH番地に格納する。1要素のサイズは配列へのポインタに7を加えたアドレスから得ることができる（バイトでリードする場合)。なお，この方法では配列の総バイト数を求めたことにならないが（なぜかは自分で考えてみてね)，誤差は最大8バイト程度なので問題はないだろう。また，サイズを求めるには配列へのポインタが指し示す「配列の総バイト数」を使えば簡単だが，ここではあえてドンくさい方法をとってしまった(反省)

なお，外部関数内でメモリをリード／ライトする場合はIOCSコールを用います。このような手順でビープ音を変更できるのですが，注意すべきことは変更したビープ音のPCMデータはX-BASICのフリーエリア内にあるということです。

ビープ音を変更したままでX-BASICを抜けるとその領域は開放されてしまいますから，ほかのプログラムを動かしているうちにそのエリアが使われて，ビープ音のPCMデータを壊してしまう恐れがあります。そうなれば，もはやまともなビープ音は鳴りません。そこで，用意したのがget_beep関数とput_beep関数です。

get_beep関数は現在のビープ音の情報(978H番地と97CH番地の内容)を退避します。put_beep関数は退避されているビープ音の情報を回復します。したがって，X-BASICに入ったときにget_beep関数を実行し，X-BASICを抜けるときにput_beep関数を実行すればビープ音を元に戻すことができます。ビープ音をあれこれと変更して遊ぶのはX-

図3　配列へのポインタが指し示すもの

リスト7　3つの外部関数定義プログラム

```
 1:          .nlist
 2: _float           equ     $0001
 3: _int             equ     $0002
 4: _char            equ     $0004
 5: _str             equ     $0008
 6: _ptr             equ     $0010
 7: _dim1_array      equ     $0020
 8: _dim2_array      equ     $0040
 9: _dimn_array      equ     $0060
10: _omit            equ     $0080
11:
12: dim1_vp          equ     $0037
13: *        =_dim1_array+_ptr+_float+_int+_char
14: int_omit         equ     $0082
15: *        =_int+_omit
16:
17: void_ret         equ     $ffff
18: int_ret          equ     $8001
19:
20: arg_cnt          equ     4
21: arg_typ          equ     6
22: arg_vec          equ     12
23:          .list
24: *********************************
25: * インフォメーション　テーブル *
26: *********************************
27:          dc.l    _ret,_ret,_ret,_ret
28:          dc.l    _ret,_ret,_ret,_ret
29:          dc.l    _token,_param,_exec
30:          dc.l    0,0,0,0,0
31: **************
32: * プログラム *
33: **************
34:
35: *****************************************
36: *        何もしない                     *
37: *****************************************
```

BASICのなかだけにしておきましょうね。以上示した，chg_beep，get_beep，put_beepという外部関数の定義をするプログラムがリスト7です。リスト1の場合と同様に，これらの外部関数をX-BASICへ取り込む方法は，バックナンバーなどを参照してください。

そしてさようなら

これまで13回に渡ってX-BASICを解説してきたこの連載も今回で一応終了です。長い間のお付き合い，本当にありがとうございました。そして，どうもお疲れさまでした。この連載の方針として，X-BASICの個々の命令の細かい使用法は意図的に避けてきました。それは命令の細部に重点を置くあまり，プログラム全体の流れがつかみにくくなるといった事態を避けたかったからです。

これはX-BASICの連載ですから，X-BASICを使ったプログラミングをいろいろと示してきましたが，X-BASICを使うということは本質ではありません。X68000を使って遊ぶということが大きな目的だったのです。そのためになにかプログラムを作る場合，いちばん大切なのはそれをどのような手順で実行するかということ，つまり，アルゴリズムです。それを実現するための言語はなんでもいいのです。ただ，X-BASICがいちばん使いやすかったのでそれを使っていただけのことです。

また，X-BASICの連載のくせにアセンブリ言語で書いた外部関数を臆面もなく載せてきたのは，アセンブリ言語とX-BASICの合体が「遊ぶ」という点でいちばん都合がよかったからにほかなりません。今回でこの連載は終了しますが，これからもX-BASIC（とアセンブリ言語）を使ってどんどん遊んでみましょう。そして，過去13カ月に渡ってご紹介してきたさまざまな事柄が，さらなる遊びのためのヒントとなれば幸いです。

最後に，この連載のタイトルは結構わけのわからないものが多かったと思いますが，これらはみな「伝説巨神イデオン」のサブタイトルのもじりなのです（第1回目だけはガンダムでしたが）。イデオンはテレビ放映のあと劇場映画として復活し，そしていまビデオでも復活しています。この連載も，果たしてイデオンよろしく，いつかまた復活することがあるのでしょうか。それは「神ならぬ身の知る由もなし」といったところです。

```
 38:         .even
 39: _ret:
 40:         rts
 41: ****************************************
 42: *        ＢＥＥＰ音の変更                *
 43: ****************************************
 44:         .even
 45: _chg_beep:
 46:         move.l  arg_vec+0(sp),d1
 47:         add.l   #10,d1              ; データの先頭
 48:         moveq.l #$88,d0             ; ロングのライト
 49:         movea.l #$000978,a1
 50:         trap    #15                 ; アドレス設定
 51:         move.w  arg_typ+10(sp),d0
 52:         addq.w  #1,d0
 53:         beq     omit_ırg            ; 引数なし
 54:         move.l  arg_vec+10(sp),d1
 55: data_set:
 56:         movea.l #$00097c,a1
 57:         moveq.l #$87,d0             ; ワードのライト
 58:         trap    #15                 ; 個数せってい
 59:         clr.l   d0                  ; エラーなし
 60:         rts
 61: omit_arg:
 62:         subq.l  #2,d1
 63:         movea.l d1,a1
 64:         moveq.l #$83,d0
 65:         trap    #15                 ; データの個数
 66:         move.w  d0,_tmp_area
 67:         subq.l  #2,d1
 68:         movea.l d1,a1
 69:         moveq.l #$83,d0
 70:         trap    #15
 71:         move.w  _tmp_area,d1
 72:         cmp.b   #1,d0               ; char か
 73:         beq     data_set
 74:         cmp.b   #4,d0               ; int か
 75:         beq     skip4
 76:         asl.w   #3,d1               ; float を仮定
 77:         bra     data_set
 78: skip4:
 79:         asl.w   #2,d1
 80:         bra     data_set
 81:         .even
 82: _get_beep:
 83:         movea.l #$000978,a1
 84:         moveq.l #$84,d0             ; アドレス
 85:         trap    #15
 86:         move.l  d0,_beep_save
 87:         move.l  d0,_ret_val
 88:         moveq.l #$83,d0             ; 個数
 89:         trap    #15
 90:         move.l  d0,_beep_save+4
 91:         clr.l   d0
 92:         lea     _ret_arg,a0
 93:         rts
 94:         .even
 95: _put_beep:
 96:         movea.l #$000978,a1
 97:         move.l  _beep_save,d1
 98:         moveq.l #$88,d0             ; アドレス
 99:         trap    #15
100:         move.l  d1,_ret_val
101:         move.l  _beep_save+4,d1
102:         moveq.l #$87,d0             ; 個数
103:         trap    #15
104:         clr.l   d0
105:         lea     _ret_arg,a0
106:         rts
107:
108:
109: ****************
110: * データエリア *
111: ****************
112:         .even
113: _token:
114:         dc.b    'chg_beep',0
115:         dc.b    'get_beep',0
116:         dc.b    'put_beep',0
117:         .even
118: _param:
119:         dc.l    _chg_beep_param
120:         dc.l    _get_beep_param
121:         dc.l    _put_beep_param
122:         .even
123: _chg_beep_param:
124:         dc.w    dim1_vp  ; 配列
125:         dc.w    int_omit ; サイズ（省略可能）
126:         dc.w    void_ret ; 戻り値（なし）
127: _get_beep_param:
128: _put_beep_param:
129:         dc.w    int_ret
130:         .even
131: _exec:
132:         dc.l    _chg_beep,_get_beep,_put_beep
133:         .even
134: _tmp_area:
135:         dc.w    0
136:         .even
137: _ret_arg:
138:         dc.w    0,0,0
139: _ret_val:
140:         dc.l    0
141:         .even
142: _beep_save:
143:         dc.l    0,0
144:         end
```

THE SENTINEL

第68部　マルチウィンドウエディタWINER

●マルチウィンドウエディタ

E-MATEに次ぐS-OS用のスクリーンエディタの登場です。すでにE-MATEの掲載された1986年5月号が品切れになり，これまで長い間E-MATEが入手できなかった方もいらっしゃるでしょう。そういう方にはまさに待望の新型スクリーンエディタの登場です。このエディタではZEDA，CAP-X85，Prolog-85，magiFORTH，FuzzyBASIC，CASL&COMET，CONTEX，STORY MASTER，tiny CORE WARS，SLANGなどのテキストデータを記述することができます。また，このエディタの呼び名ですが“ワイナー”ではなく“ウィナー”というふうに読んでくださいね。

さて，E-MATEの強化版として送られてくる投稿のほとんどはコントロールキー対応にしたり，単にキーアサインを変更したり，高速化したりといった機種ごとの特殊化を行ったものでした。もちろん，掲載されたプログラムを各自で機能強化するというのはS-OSの正しい使い方なのですが，新しい発想のエディタが現れないのもちょっと問題です。

新しいエディタを作る際の最大の問題となっているのはS-OSにコントロールキーがないということでしょう。しかし，すべてのマシンにコントロールキーやファンクションキーがあるわけではありません。それでもすべてのマシンでなんらかのエディタは走っています。エディタにはコントロールキーが必須という固定観念をなくすことが重要でしょう。UNIXなどでは，端末を選ばないソフトウェアの書き方がある程度確立されています。

E-MATEのようにコントロールモードを設定してしまうか，このWINERのようにコントロールコードをエスケープシーケンスにしてしまうか。viのようにコマンドモードだけで間に合わせてしまう手もありますし，いろいろと方法はあると思います。こういったところをクリアしたうえで，各機種にコントロールキー対応のモードもつけるといったかたちのものが望ましいでしょう。

WINERの作者である近藤さんはさらにWINERのパワーアップを計画しているようです。これも近いうちに発表できるでしょう。また，残念ながら今月は各機種用のラインプリントルーチンを載せるスペースが確保できませんでした。とりあえず，共通ルーチンで使用してみてください（もちろん，共通ルーチンを参考にして各機種用のルーチンを自作されても結構ですが）。

全機種共通システム掲載記事

■85年6月号
序論　共通化の試み
第1部　S-OS"MACE"
第2部　Lisp-85インタプリタ
第3部　チェックサムプログラム
■85年7月号
第4部　マシン語プログラム開発入門
第5部　エディタアセンブラZEDA
第6部　デバッグツールZAID
■85年8月号
第7部　ゲーム開発パッケージBEMS
第8部　ソースジェネレータZING
■85年9月号
インタラプト　S-OS番外地
第9部　マシン語入力ツールMACINTO-S
第10部　Lisp-85入門(1)
■85年10月号
第11部　仮想マシンCAP-X85
連載　Lisp-85入門(2)
■85年11月号
連載　Lisp-85入門(3)
■85年12月号
第12部　Prolog-85発表
■86年1月号
第13部　リロケータブルのお話
第14部　FM音源サウンドエディタ
■86年2月号
第15部　S-OS"SWORD"
第16部　Prolog-85入門(1)
■86年3月号
第17部　magiFORTH発表
連載　Prolog-85入門(2)
■86年4月号
第18部　思考ゲームJEWEL
第19部　LIFE GAME
連載　基礎からのmagiFORTH
連載　Prolog-85入門(3)
■86年5月号
第20部　スクリーンエディタE-MATE
連載　実戦演習magiFORTH
■86年6月号
第21部　Z80TRACER
第22部　magiFORTH TRACER
第23部　ディスクダンプ&エディタ
第24部　"SWORD"2000 QD
連載　対話で学ぶmagiFORTH
特別付録　PC-8801版S-OS"SWORD"
■86年7月号
第25部　FM音源ミュージックシステム
付録　FM音源ボードの製作
連載　計算力アップのmagiFORTH
特別付録　SMC-777版S-OS"SWORD"
■86年8月号
第26部　対局五目並べ
第27部　MZ-2500版S-OS"SWORD"
■86年9月号
第28部　FuzzyBASIC発表
連載　明日に向かってmagiFORTH
■86年10月号
第29部　ちょっと便利な拡張プログラム
第30部　ディスクモニタDREAM
第31部　FuzzyBASIC料理法<1>
■86年11月号
第32部　パズルゲームHOTTAN
第33部　MAZE in MAZE
連載　FuzzyBASIC料理法<2>
■86年12月号
第34部　CASL & COMET
連載　FuzzyBASIC料理法<3>
■87年1月号
第35部　マシン語入力ツールMACINTO-C
連載　FuzzyBASIC料理法<4>
■87年2月号
第36部　アドベンチャーゲームMARMALADE
第37部　テキアベ作成ツールCONTEX
■87年3月号
第38部　魔法使いはアニメがお好き
第39部　アニメーションツールMAGE
付録　"SWORD"再掲載とMAGICの標準化
■87年4月号
第40部　INVADER GAME
第41部　TANGERINE
■87年5月号
第42部　S-OS"SWORD"変身セット
第43部　MZ-700用"SWORD"をQD対応に
■87年6月号
インタラプト　コンパイラ物語
第44部　FuzzyBASICコンパイラ
第45部　エディタアセンブラZEDA-3
■87年7月号
第46部　STORY MASTER
■87年8月号
第47部　パズルゲーム碁石拾い
第48部　漢字出力パッケージJACKWRITE
特別付録　FM-7/77版S-OS"SWORD"
■87年9月号
第49部　リロケータブル逆アセンブラInside-R
特別付録　PC-8001/8801版S-OS"SWORD"
■87年10月号
第50部　tiny CORE WARS
第51部　FuzzyBASICコンパイラの拡張
第52部　X1turbo版S-OS"SWORD"
■87年11月号
序論　神話のなかのマイクロコンピュータ
付録　S-OSの仲間たち
第53部　もうひとつのFuzzyBASIC入門
第54部　ファイルアロケータ&ローダ
インタラプト　S-OSこちら集中治療室
第55部　BACK GAMMON
■87年12月号
第56部　タートルグラフィックパッケージTURTLE
第57部　X1turbo版"SWORD"アフターケア
ラインプリントルーチン
特別付録　PASOPIA7版S-OS"SWORD"
■88年1月号
第58部　Fuzzy BASICコンパイラ・奥村版
付録　石上版コンパイラ拡張部の修正
■88年2月号
第59部　シューティングゲームELFES
■88年3月号
第60部　構造型コンパイラ言語SLANG
■88年4月号
第61部　デバッギングツールTRADE
第62部　シミュレーションウォーゲームWALRUS
■88年5月号
第63部　シューティングゲームELFES II
第64部　地底最大の作戦
■88年6月号
第65部　構造化言語SLANG入門(1)
第66部　Lisp-85用NAMPAシミュレーション
■88年7月号
第67部　マルチウィンドウドライバMW-1
連載　構造化言語SLANG入門(2)
*以上のアプリケーションは，基本システムであるS-OS"MACE"またはS-OS"SWORD"がないと動作しませんので ご注意ください。

全機種共通S-OS“SWORD”要

マルチウィンドウエディタWINER

お待ちかね，S-OS用スクリーンエディタ第2弾の登場です。エディタはS-OSでのプログラム開発を多方面から支援してくれる便利なユーティリティですので，皆さんぜひ打ち込んで活用してください。来月は各機種用にパワーアップを予定しています。

近藤　環 *Kondou Tamaki*

マルチウィンドウエディタ

このエディタではX68000のWINDEXのようなマルチウィンドウ環境を実現します。オーバーラッピングマルチウィンドウをサポートしていますが，7月号で発表されたマルチウィンドウパッケージMW-1とは特に関係ありません。MW-1のウィンドウは階層構造になっていますので，ウィンドウ間の自由な移動が必要なウィンドウエディタなどは作りづらかったのですが，このWINERがあれば問題はないでしょう。

WINERで使われているウィンドウシステムは汎用のものではありません。ゲームやユーティリティではむしろMW-1のようなシステムのほうが向いているでしょう。このシステムではウィンドウはウィンドウの移動などの際に下から書き直されます。これにより，画面情報の退避エリアや仮想画面といったメモリを多量に消費するものを使わずにマルチウィンドウを実現しています。

また，操作法については，S-OSの標準的なスクリーンエディタであるE-MATEにほぼ準じていますし，機能的にもマクロ機能を取り入れるなど意欲的な拡張が見られます。

入力方法

特殊ワークを4Kバイト使用しますので，MZ-80Kなどでは特殊ワークの拡大が必要です（すでにZEDA3などで特殊ワークを拡大している場合には不要）。

リスト1は全機種共通ですので，そのまま各機種のマシン語モニタやMACINTO-Cなどのマシン語入力ツールから打ち込んでください。デバイスにセーブしチェックサム，CRCチェックバイトなどを確認したら，

#J3000

でWINERを起動してください。

使用方法

基本的なキー操作はE-MATEに準じています。コントロールキーはブレイク＋各キーに割り当てられています。S-OSのブレイクコードは1BHですから，ESCキーをそのまま通している機種では，コントロールキーがエスケープシーケンスに置き換えられているわけです。以下，CTRL＋～という表現はブレイクキーまたはESCキーに続いてなにかのキーを押すことを表しています。コントロールキーのついている機種では直接コントロールキーで操作できるように“SWORD”内のキャラクタテーブルを変更しておくのもよいでしょう。具体的なコントロールコードは表1にまとめてあります。

E-MATEと違い，1文字でコントロールモードから抜けてしまいますが，CTRL＋スペースでE-MATEのコントロールモードのようなコントロールロック状態になります（抜けるときはもう一度スペースキーを押す）。

ほとんどのコントロールコードはどのウィンドウに対しても有効ですが，起動時に開かれているウィンドウ番号のついていないウィンドウ（コマンドウィンドウ）ではWINER特有のコマンドを実行することができます。これらは表2にまとめておきます。

●なにも考えずに使う

WINERを当たり前のスクリーンエディタとして使用する場合は以下のようにしてください。ウィンドウを使用しないのであれば，これらの操作を最初からワークエリアに書き込んでおくとよいでしょう。

1)　M/B：adr

adrは各機種のMEMAXのアドレス。起動後，Mコマンドでテキストエリア Aの割り当てメモリを最大にする（巨大なテキストでなければ，この操作は不要）。

2)　L

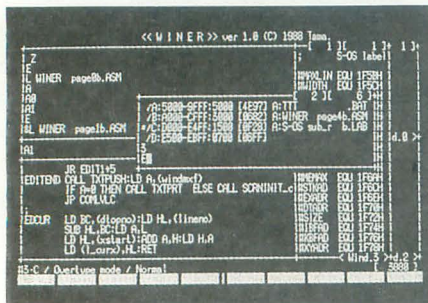
ディレクトリを表示させ，テキストを読み込む（新しいテキストでは不要）。

3)　0E

ウィンドウ0のエディットを開始する。

4)　CTRL＋”

ウィンドウ0を画面いっぱいに拡大し，エディット開始する。

図1　メモリマップ

●**マルチウィンドウで使う**

複数のウィンドウで同一のテキスト，複数のテキストを扱うことができます。同一のテキストを複数のウィンドウで扱う場合，画面上のテキスト書き換えはアクティブウィンドウが変わったとき，ウィンドウの位置や大きさを切り換えたときなどに更新されます。

まず，コマンドウィンドウからアクティブウィンドウを指定しましょう。標準状態では0番のウィンドウがアクティブになっていますので，ウィンドウ0をアクセスしたければそのまま，それ以外のウィンドウをアクセスしたければ，該当するウィンドウ番号（0～7）を入力してください。これによって任意のウィンドウを呼び出すことができます。エディットを開始するときはEコマンドを打ち込みます。まとめて，

1E

のように入力してもかまいません。

エディット状態からアクティブウィンドウを切り換えるには，CTRL+！を使用します。画面の指示に従って切り換えるウィンドウ番号を入力してください。

このWINERではMW-1のような画面情報の保護/復帰というものは行わず，ウィンドウを下から順に書き直すことで画面を保っています。そのため，表示速度の遅い機種では少しイラつくこともあるかもしれません。そこで，下画面の不要なときは画面の書き換えを行わないというモードをつ

表1　コントロールキー一覧

コントロールキーの実行方法

[ESC]または[SHIFT]+[BREAK]を押しコントロールモードに入り，次に任意のコントロールキーを押すことで実行される。実行したら通常の入力モードに戻る。コントロールモードの状態は最下段に表示が出る。

カーソル移動キー

キー	コード	機能
^ S ←	013H 01DH	1文字左に移動
^ D →	004H 01CH	1文字右に移動
^ E ↑	005H 01EH	1文字上に移動
^ X ↓	018H 01FH	1文字下に移動
^ A	001H	1単語左に移動
^ F	006H	1単語右に移動

画面表示関係

キー	コード	機能	
^ W	017H	1行下にスクロール	*
^ Z	01AH	1行上にスクロール	*
^ R	012H	1ページ下にスクロール	*
^ C	003H	1ページ上にスクロール	*
^ J	00AH	画面を表示しなおす	*
^ T	013H	1ステップ左にスクロール	*
^ U	014H	1ステップ右にスクロール	*

タビュレーション関係

キー	コード	機能
^ I	009H	次のタブ位置までカーソルを移動する
^ Q	011H	タブ位置の設定，解除を行う

ウィンドウコントロール関係

キー	コード	機能	
^ !	021H	ダイレクトに指定のウィンドウにジャンプする	*
^ "	022H	現在使用中のウィンドウの大きさを最大にし，もう一度押すともとの大きさに戻る	
^ #	023H	ウィンドウの移動。カーソルがウィンドウの左上に移動しテンキー，カーソルキー，Sを中心としたキーでカーソルを動かし，上記以外のキーを押すと移動したウィンドウを表示する	
^ $	024H	ウィンドウの大きさ指定。同上	

コピー関係

キー	コード	機能
^ 0	030H	コピーバッファを表示する。スペースキーを押すたびにCH1→CH2→CH3の順で表示される
^ 1	031H	コピーバッファ1の内容を出力する
^ 2	032H	コピーバッファ2の内容を出力する
^ 3	033H	コピーバッファ3の内容を出力する
^ 4	034H	コピーバッファ1の内容を消去する
^ 5	035H	コピーバッファ2の内容を消去する
^ 6	036H	コピーバッファ3の内容を消去する
^ 7	037H	カーソルの文字をコピーバッファ1に入れる
^ 8	038H	カーソルの文字をコピーバッファ2に入れる
^ 9	039H	カーソルの文字をコピーバッファ3に入れる

デリート関係（削除した文字はデリートバッファに保存）

キー	コード	機能	
^ G	007H	カーソル上の文字を削除する	
^ B	002H	カーソルの左の文字を削除する	
^ V	016H	カーソル以後1行を削除する	
^ Y	019H	改行コードを含む1行を削除する	*

マクロ関係

キー	コード	機能
^ ;	03BH	マクロ0実行
^ :	03AH	マクロ1実行
^ ,	02CH	マクロ2実行
^ .	02EH	マクロ3実行
^ ／	02FH	マクロ4実行
^ ＋	02BH	マクロ5実行
^ ＊	02AH	マクロ6実行
^ ＜	03CH	マクロ7実行
^ ＞	03EH	マクロ8実行
^ ?	03FH	マクロ9実行

カット&ペースト関係

キー	コード	機能	
^ %	025H	カーソルの行にカット&ペースト用のマークをつける。マークがついているときはウィンドウの右上の行No.の表示部に＊がつく	*
^ &	026H	マークした行からカーソルまでの行をバッファに入れテキストを削除する	*
^ '	027H	マークした行からカーソルまでの行をバッファに入れテキストを削除しない	*
^ (	028H	カーソルの行からバッファの内容をインサートする	*

行コネクト関係

キー	コード	機能	
^ N	00EH	カーソルの位置でその行を分割する	*
^ L	00CH	カーソルの行と次の行を1文字あけて連結しカーソルを連結部に移動する	*

デリートバッファ関係

キー	コード	機能
^ P	010H	カーソルの位置からデリートバッファの内容を出力する
^ K	00BH	デリートバッファの内容をクリアする

その他

キー	コード	機能	
^ O	00FH	オーバーライトモードとインサートモードの切り換え	
^SP	020H	コントロールモードキープON/OFFの切り換え	
^ H	008H	現在の行を変更前に戻す	
^ESC	01BH	エディットモードからコマンドレベルに戻る（再設定不可）	*
^ M RET	00DH 00DH	改行する	

＊印はコマンドウィンドウで使用不可
なお，ESCはコントロールモードに入るときの操作と同様の操作を表す

THE SENTINEL

けてあります。これはコマンドウィンドウの上からPコマンドを実行するたび，モードが切り換わっていきます（標準状態では書き換えは行わない）。

●ファイルをロード/セーブする

ファイルのロード/セーブはコマンドウィンドウ上から行います。これらのコマンドは現在選択されているウィンドウに対して有効になりますから，Lコマンドで読み込んでください。ファイル名が省略されたり，不適当な場合はディレクトリが表示されますので，カーソルをロードしたいファイルのところまで運んでリターンキーを押してください。

WINERはマルチテキストですので，テキスト編集中にちょっとほかのテキストを参照したいといった程度のことならばいちいちファイルをロード/セーブすることもありません。

各ウィンドウからはCTRL+CTRLでコマンドウィンドウにジャンプできますから，エディットが終了した時点でコマンドウィンドウに移りSコマンドを使ってセーブしてください。Lコマンドで読み込んだものについてはファイル名を覚えていますのでいちいち指定する必要はありません。

●テキストをコピーする

コピー機能関係はCTRL＋0～9の数字キーに割り振られています。もっとも基本的な使用方法は，

1) コピーしたい文字列の先頭にカーソルをあわせる。
2) CTRL＋スペースでコントロールモードをロックする。
3) 数字の7のキーを押してバッファに文字列を送る。
4) 転送したい場所にカーソルをあわせ，数字の1のキーを押す。
5) 4のキーでバッファの内容をクリアしておく。

という手順になります。

この例ではバッファ1しか使っていませんが，テキストのコピーバッファには3種類があります。テンキーのある機種では縦一列にバッファ1，2，3が並んでいますのでわかりやすいでしょう。

表2 各コマンドの説明

n……行No.1～65534　fn……ファイルネーム　t……テキストエリア名A～D
str……文字列　(　)……省略できる

コマンド	説明
E(n)	エディットモードに入る
0～7	ウィンドウを選択する。続けてコマンドを入力できる
/t	テキストエリアを選択する
L(fn)	現在のテキストエリアにテキストをロードする ファイル名を省略するとディレクトリを表示して，その中からカーソルで選択する
S(fn)	現在のテキストエリアをセーブする ファイル名を省略すると，テキストが一度ロードしたものであれば，ファイル名を表示してそのままセーブするかどうか聞いてくるので，Yesならば，リターンキーを押す。別のキーを押すか新規のファイルであればLコマンドと同じ
Dn n	現在のテキストエリアの指定した範囲を削除する
B	ラインNo.をトップにする。F,Cコマンドで使用する
F(n)" str"	現在のテキストエリアで文字列を検索する。文字列をみつけると文字列の最初でカーソルキーが点滅する。そのときスペースキーを押すとエディットモードに入る
C(/)(n)" str1 "str2"	現在のテキストエリアの"str1"を"str2"に置き換える /をつけると"str1"をみつけるたび文字列の最初でカーソルが点滅する。そのとき変換したかったらスペースキーを押す
M	現在のテキストエリアの使用状況を表示する たとえば/A：5000―8000：3000［2FFF］と表示されたらテキストエリアAは5000Hから8000Hまでの3000Hバイトあり，残り2FFFHバイトある
M/t：adr	テキストエリアの開始アドレスを指定し使用状況を表示する
N	現在のウィンドウの割り当てを表示する たとえば0/A：なら，ウィンドウ0にはテキストエリアAが割り当てられている
Xt t	テキストエリア内の交換
P	ウィンドウをプライオリティに合わせて表示するかどうかの切り換え
%	タブデータを表示するかどうかの切り換え
&	現在のテキストエリアのテキストを消去する
R	&コマンドで消したテキストを復活する
!	呼び出したシステムに帰る

より高度な使い方

以上がWINERの基本的な使い方ですが，設定を変えたり，専用ルーチンを組み込むことでさらに自分にあった使い方ができるようになります。

●スクロールを高速にしたい

WINERには専用のラインプリントルーチンを組み込むことができます(E-MATEのものは使えません)。今月は残念ながら各機種用ラインプリントルーチンを掲載することはできませんでした。来月にご期待ください。

●複数のテキストを同時に扱いたい

WINERはマルチテキストに対応しています。扱えるテキストは4種類でA～Dの名前で識別されます。テキストの割り当てを行ってみましょう。0～3のウィンドウにA～Dのテキストを割り当てる場合，

0/A
1/B
2/C
3/D

のようにコマンドウィンドウから打ち込めばよいのです。テキストの割り当てを行わなければ，どのウィンドウもAのテキストをエディットすることになります。

●もっと大きなテキストを扱いたい

標準の状態ではマルチテキストモードですので，

A　5000H～9FFFH
B　A000H～CFFFH
C　D000H～E4FFH
D　～MEMAX

というテキストエリアの割り当てが行われています。もっと大きなテキストを扱いたいときには，これを変更してシングルテキストにすればよいのです。具体的にテキストAを大きくするためには，コマンドウィンドウから，

M/B：C000
M/C：……

のようにしてテキストB以降の先頭アドレスを変えます。

テキストエリアを最大にしてもE-MATEより若干フリーエリアは狭くなりますが，これはいたしかたないでしょう。

●面倒なキー操作をまとめて行いたい

テキストDは特殊な用途に使えます。すなわち，この領域に書いたテキストはマクロテキストとみなされ，コントロールキーでバッチファイルのように自動実行することができるのです。具体的な書式を示して

おきましょう。

Dのテキストに次のように書き込んでみます。

```
MCR1
0/A@M
L TEST1@M
1/B@M
L TEST2@M
P@M
E@M
@"@$888888888888@M
@#222222222222@M
@!0
@"@$22222222222@M
END
```

1行目はマクロ番号1番の命令を記述していることを示します。ENDまでの記述はキー操作を表し，コントロールコードは@を頭につけて表現します。操作内容についてはコントロールコード表から解読してください。

さて，ここでコマンドウィンドウからCTRL+：(コロン)を実行してみると，ウィンドウ0にTEST1，ウィンドウ1にTEST2というファイルを読み込んで，上下半分ずつのウィンドウを開きエディット状態に入ります。

このようなマクロ機能は0～9番までの10種類が登録可能で，定型的なテキスト編集作業には威力を発揮します。

Profile

◇近藤さんは大阪府にお住まいの24歳。塾の講師をされているそうです。パソコン歴は約4年のX1turboユーザー，今度はカルクやデータベースにも挑戦したいとか。

表3　各種ワークの説明

メインメモリ側

300E$_H$　(4E00$_H$)　タブデータバッファアドレス
3010$_H$　(4F00$_H$)　1行バッファアドレス
3012$_H$　(5000$_H$)　テキストA開始アドレス
3014$_H$　テキストB開始アドレス
3016$_H$　テキストC開始アドレス
3018$_H$　テキストD開始アドレス
301A$_H$　テキストMAX(MEMAX)
テキストエリアの設定はMコマンドでもできる

特殊ワーク側

301C$_H$　(0000$_H$)　カット＆ペースト用バッファアドレス（0000$_H$に固定する）
301E$_H$　(0700$_H$)　コピー用バッファドレス(同時にC＆Pバッファの上限になる)
3020$_H$　(0A00$_H$)　ウィンドウデータ用ワークアドレス
3022$_H$　(0B00$_H$)　サーチ用文字列バッファアドレス
3024$_H$　(0B80$_H$)　チェンジ用文字列バッファアドレス
3026$_H$　(0C00$_H$)　コマンドテキスト用バッファアドレス
3028$_H$　(07$_H$)　コマンドテキスト用ラインマスク
＊ラインNo.にかかるマスクで1行は128バイトなので使えるのは
03$_H$―　3行　大きさは0200$_H$
07$_H$―　7行　大きさは0400$_H$
0F$_H$―　15行　大きさは0800$_H$
1F$_H$―　31行　大きさは1000$_H$
3F$_H$―　63行　大きさは2000$_H$
7F$_H$―127行　大きさは4000$_H$
FFH―255行　大きさは8000$_H$
3029$_H$　(0000$_H$)　デリートバッファ用アドレス
＊初期値は0000$_H$でC＆Pバッファと同じだがコマンドテキストバッファの後ろにすれば重ならないで使える
302B$_H$　(03$_H$)　デリートバッファ用サイズマスク
＊アドレスの上位8ビットにかかるマスクなので使えるのは
01$_H$―大きさは0200$_H$
03$_H$―大きさは0400$_H$
07$_H$―大きさは0800$_H$
0F$_H$―大きさは1000$_H$
1F$_H$―大きさは2000$_H$
3F$_H$―大きさは4000$_H$
7F$_H$―大きさは8000$_H$
たとえばX1turboなら特殊ワークの設定を頭から
0000$_H$, 8000$_H$, 8300$_H$, 8400$_H$, 8480$_H$, 8500$_H$, 3F$_H$, A500$_H$, 0F$_H$とすると，
カット＆ペーストバッファが32Kバイト，コマンドテキストが8Kバイト(63行)，
デリートバッファが4Kバイトとなる

リスト1　WINERダンプリスト

```
3000 CD F1 31 CD 0C 32 C3 50 : 0D
3008 32 00 00 00 FE EC 00 4E : 6A
3010 00 4F 00 50 00 A0 00 D0 : 0F
3018 00 E5 00 EC 00 00 00 07 : D8
3020 00 0A 00 0B 80 0B 00 0C : AC
3028 07 00 00 03 18 0D 00 50 : 7F
3030 17 00 18 07 2E 06 03 02 : 6F
3038 36 11 08 03 36 11 0D 04 : AA
3040 36 11 12 05 36 11 01 02 : A8
3048 2A 0C 2D 02 22 0C 01 10 : A4
3050 2A 06 2D 10 22 06 18 07 : B4
3058 2E 06 06 03 00 00 00 00 : 3D
3060 50 00 00 00 50 00 50 00 : F0
3068 50 00 A0 00 0D 00 00 4D : 4A
3070 00 00 00 01 00 00 00 00 : 01
3078 00 00 FF FF CA E5 00 01 : AE
---------------------------------
SUM: AB 69 62 3B A7 F5 3D 3E 95ED

3080 0D 00 01 02 03 04 05 06 : 22
3088 07 08 09 0A 0B 0C 0D 0E : 54
3090 0F 10 11 12 13 14 15 16 : 94
3098 17 18 19 1A 1B 1C 1D 1E : D4
30A0 1F 20 21 22 23 24 25 26 : 14
30A8 27 28 29 2A 2B 2C 2D 2E : 54
30B0 2F 30 31 32 33 34 35 36 : 94
30B8 37 38 39 3A 3B 3C 3D 3E : D4
30C0 3F 20 3A 3B 2C 3D 2B 2D : 95
30C8 2A 2F 3C 3E 28 29 5B 5D : DC
30D0 7B 7D A2 A3 22 27 20 20 : C6
30D8 20 01 00 FF FF FF FF FF : 1C
30E0 FF FF FF 3C 3C 20 57 20 : 0C
30E8 49 20 4E 20 45 20 52 20 : AE
30F0 3E 3E 20 76 65 72 20 31 : 3A
30F8 2E 30 20 28 43 29 20 31 : 63
---------------------------------
SUM: 9E 3A 8D 05 96 67 96 5B B521

3100 39 38 38 20 54 61 6D 61 : 4C
3108 2E 00 20 20 20 20 32 00 : E0
3110 5B 20 20 20 20 20 5D 5B : B3
3118 20 20 20 20 20 20 20 5D : 3D
3120 00 20 2F 20 49 6E 73 65 : FE
3128 72 74 20 6D 6F 64 65 20 : CB
3130 20 20 2F 20 00 20 2F 20 : FE
3138 4F 76 65 72 74 79 70 65 : 5E
3140 20 6D 6F 64 65 20 2F 20 : 34
3148 00 4E 6F 72 6D 61 6C 00 : 69
3150 43 74 72 6C 20 00 3C 20 : 11
3158 57 69 6E 64 2E 00 48 69 : 71
3160 74 20 3C 43 52 3E 20 6F : 32
3168 6E 20 46 69 6C 65 20 6F : 9D
3170 72 20 42 72 65 61 6B 1E : 95
3178 0D 00 57 68 69 63 68 20 : 20
---------------------------------
SUM: DE 9A 54 CB 8C 14 C5 E8 7CB0

3180 77 69 6E 64 6F 77 20 64 : 1C
3188 6F 20 79 6F 75 20 6A 75 : EB
3190 6D 70 20 3F 20 20 20 00 : 9C
3198 4E 6F 20 75 73 69 6E 67 : 03
31A0 00 4E 6F 77 20 6C 6F 61 : 90
31A8 64 69 6E 67 20 00 20 44 : 26
31B0 6F 20 79 6F 75 20 73 61 : E0
31B8 76 65 20 3F 20 00 4E 6F : 17
31C0 77 20 73 61 76 69 6E 67 : 1F
31C8 20 20 00 4D 65 6D 6F 72 : 40
31D0 79 20 4F 76 65 72 20 20 : 75
31D8 20 20 20 20 20 20 20 00 : E0
31E0 20 50 72 77 20 6F 6E 00 : 56
31E8 20 50 72 77 20 6F 66 66 : B4
31F0 00 CD 10 34 06 04 11 6F : 9B
31F8 30 78 3D D5 C5 CD CC 34 : 4C
---------------------------------
SUM: 8A 09 B0 4E B7 C3 36 B7 FFF4

3200 C1 D1 38 04 7E 36 00 12 : 94
3208 1B 10 EE C9 2A 20 30 AF : 0B
3210 47 CD 9A 1F 23 10 FA AF : A9
3218 32 31 30 21 00 00 CD 9A : 1B
3220 1F 2A 1E 30 CD 9A 1F 24 : 41
3228 CD 9A 1F 24 CD 9A 1F 21 : 51
3230 00 00 22 78 30 CD CC 34 : 97
3238 22 67 30 ED 53 69 30 CD : 5F
3240 EC 34 D0 2A 1A 30 22 18 : 9E
3248 30 22 16 30 22 14 30 C9 : C7
3250 ED 73 0C 30 AF 67 6F 32 : 53
3258 5B 30 32 5A 30 22 70 30 : 09
3260 22 73 30 22 7F 30 2A 26 : E6
3268 30 11 80 00 3A 28 30 47 : 9A
3270 04 AF CD 9A 1F 19 10 FA : 5C
3278 CD FC 33 CD 44 33 CD 52 : 5F
---------------------------------
SUM: EA 32 53 33 1F 41 99 4C 2E5D

3280 33 C3 39 41 AF 32 75 30 : F6
3288 3A 31 30 FE 08 38 01 AF : 89
3290 32 31 30 87 87 21 36 30 : 28
3298 85 6F 11 32 30 01 04 00 : 6C
32A0 ED B0 3A 31 30 6F 87 85 : B3
32A8 2A 20 30 85 6F 30 01 24 : C3
32B0 EB 21 5C 30 01 03 00 CD : 69
32B8 91 1F 3A 5C 30 CD CC 34 : 43
32C0 30 06 CD C4 1F C3 81 32 : 5C
32C8 22 67 30 ED 53 69 30 2A : BC
```

THE SENTINEL

```
32D0 34 30 ED 5B 32 30 ED 4B : 46
32D8 2F 30 7D FE 10 30 02 2E : 4A
32E0 10 24 25 20 01 24 1C 1D : D7
32E8 20 01 1C 14 15 20 01 14 : 9B
32F0 7B 85 B9 38 04 1D 20 F8 : 2A
32F8 1C 7B 85 B9 38 03 2D 18 : 55
---------------------------------
SUM: 33 96 90 69 44 EB 0E CF D0D6

3300 F8 7A 84 B8 38 04 15 20 : 1F
3308 F8 14 7A 84 B8 38 03 25 : 22
3310 18 F8 22 34 30 ED 53 32 : 08
3318 30 C9 3A 31 30 87 87 11 : B3
3320 36 30 83 5F 21 32 30 01 : CC
3328 04 00 ED B0 3A 31 30 6F : AB
3330 87 85 2A 20 30 85 6F 30 : AA
3338 01 24 11 5C 30 01 03 00 : C6
3340 EB C3 97 1F 3A 5C 1F 32 : 4B
3348 2F 30 3A 5B 1F 3D 32 30 : B2
3350 30 C9 3E 0C CD F4 1F 3A : 5D
3358 5C 1F 21 18 00 FE 50 28 : 2A
3360 02 2E 01 CD 1E 20 11 E3 : 30
3368 30 CD E5 1F 3A D9 30 B7 : FB
3370 28 15 CD D8 33 11 DA 30 : 30
3378 1A 3C C8 D5 3D 32 31 30 : C3
---------------------------------
SUM: 14 4F B0 63 F9 60 D0 E6 AAE0

3380 CD 87 33 D1 13 18 F1 CD : 41
3388 84 32 CD 3B 34 CD 10 38 : 07
3390 AF 32 2E 30 32 7E 30 2A : 49
3398 32 30 ED 4B 34 30 CD 00 : CB
33A0 4C CD EB 37 CD B5 37 CD : C1
33A8 D1 37 CD A9 34 2A 32 30 : 3E
33B0 22 2C 30 3A 35 30 47 ED : 51
33B8 5B 63 30 21 2D 30 1A B7 : 3D
33C0 28 0A CD 03 4C CD 76 34 : C5
33C8 34 10 F3 C9 3E 0D 12 CD : 2A
33D0 03 4C 34 10 FA AF 12 C9 : 17
33D8 3A 31 30 5F 01 FF 08 21 : 23
33E0 DA 30 7B BE 28 0A 79 BE : AC
33E8 28 0C 23 10 F5 C3 81 32 : D2
33F0 23 7E 2B B9 20 02 73 C9 : E3
33F8 77 23 18 F4 21 DB 30 54 : 26
---------------------------------
SUM: 01 22 38 78 F3 04 07 C8 6FCB

3400 5D 13 01 07 00 36 FF ED : 9A
3408 B0 3A 31 30 32 DA 30 C9 : 50
3410 2A 6A 1F 22 1A 30 06 04 : 29
3418 DD 21 18 30 DD 5E 00 DD : 5E
3420 56 01 DD 6E 02 DD 66 03 : EA
3428 B7 ED 52 30 07 19 DD 75 : 98
3430 00 DD 74 01 DD 2B DD 2B : 62
3438 10 E2 C9 2A 5D 30 CD 04 : 43
3440 35 22 5F 30 ED 43 5D 30 : A3
3448 EB 60 69 3A 35 30 CB 3F : 5D
3450 4F 47 04 18 03 CD 80 34 : 36
3458 10 FB B7 ED 42 30 03 21 : 45
3460 00 00 22 61 30 2A 67 30 : 74
3468 B7 ED 52 38 04 ED 5B 67 : E1
3470 30 ED 53 63 30 C9 1A B7 : 9D
3478 C8 13 FE 0D 20 F8 B7 C9 : 7E
---------------------------------
SUM: 5F 36 1D CA 57 37 60 19 1115

3480 1B 1B 1A B7 28 04 FE 0D : 3E
3488 20 F7 13 C9 2A 0E 30 7E : D9
3490 FE 2D C8 FE 2B C8 0E 20 : 12
3498 36 2B 23 06 07 36 2D 23 : 17
34A0 10 FB 0D 20 F3 2B 36 0D : 99
34A8 C9 CD 8C 34 3A 6B 30 B7 : E2
34B0 C8 2A 2C 30 E5 2A 32 30 : BF
34B8 3A 35 30 84 67 22 2C 30 : 08
34C0 ED 5B 0E 30 CD 03 4C E1 : 83
34C8 22 2C 30 C9 FE 04 3F D8 : 60
34D0 21 12 30 87 85 6F 5E 23 : 5F
34D8 56 23 7E 23 66 6F B7 ED : 93
34E0 52 F5 44 4D 19 F1 EB D8 : A5
34E8 37 C8 B7 C9 ED 4B 67 30 : 4E
34F0 C5 2A 69 30 B7 ED 42 44 : B2
34F8 4D E1 AF ED B1 2B 22 65 : 2D
---------------------------------
SUM: 6B 15 0C 62 21 2B 83 6C D53E

3500 30 C8 37 C9 D5 EB 2A 67 : 49
3508 30 01 00 00 7A B3 28 0F : 95
3510 7E B7 28 0D 23 FE 0D 20 : B8
3518 F7 03 1B 7A B3 20 F1 D1 : 24
3520 C9 22 65 30 18 F9 ED 5B : D9
3528 5F 30 2A 10 30 06 FF 1A : 18
3530 FE 20 38 0C 77 13 23 10 : 1F
3538 F6 18 0A 2A 10 30 06 FF : 87
3540 36 20 23 10 FB 36 0D C9 : 90
3548 3A 7F 30 B7 C8 CD D1 35 : 3B
3550 60 69 E5 ED 5B 5F 30 1A : 9F
3558 B7 20 0F C1 CD BB 35 D8 : 3C
3560 CD 73 35 AF 12 ED 53 65 : DB
3568 30 C9 CD C6 35 B7 ED 42 : A7
3570 20 0C C1 2A 10 30 ED 5B : 9F
3578 5F 30 ED B0 B7 C9 30 18 : F4
---------------------------------
SUM: F4 AD 42 8A ED B8 05 F5 6910

3580 19 EB E5 44 4D 2A 65 30 : 39
3588 B7 ED 42 44 4D 03 E1 ED : 48
3590 B0 1B ED 53 65 30 18 DA : 92
3598 44 4D CD BB 35 30 02 C1 : 41
35A0 C9 ED 4B 65 30 09 E5 60 : E4
35A8 69 B7 ED 52 44 4D 03 D1 : C4
35B0 2A 65 30 ED 53 65 30 ED : 81
35B8 B8 18 B7 E5 D5 CD 06 36 : 4A
35C0 AF ED 42 D1 E1 C9 01 00 : 5A
35C8 00 1A 03 13 FE 0D 20 F9 : 54
35D0 C9 01 00 01 2A 10 30 09 : 3E
35D8 2B 2B 7E FE 20 20 06 0B : 23
35E0 78 B1 20 F5 03 23 36 0D : A7
35E8 C9 1A FE 20 C0 13 18 F9 : E5
35F0 FE 61 D8 FE 7B D0 D6 20 : 76
35F8 C9 E5 C5 21 C1 30 01 18 : 9E
---------------------------------
SUM: 83 A5 7E 36 F8 51 FA 57 A069

3600 00 ED B1 C1 E1 C9 2A 69 : 9C
3608 30 ED 5B 65 30 13 B7 ED : C4
3610 52 C9 CD 1B 36 7C B5 28 : 92
3618 01 2B C9 21 00 00 1A CD : FD
3620 32 36 D8 29 44 4D 29 29 : 4C
3628 09 D6 30 4F 06 00 09 13 : 80
3630 18 EC FE 30 D8 FE 3A 3F : 81
3638 C9 23 7C B5 20 01 2B 01 : 6A
3640 0F 31 AF 02 0B 11 0A 00 : 17
3648 CD 61 36 7B C6 30 02 0B : E2
3650 7D B4 20 F1 21 09 31 B7 : 54
3658 ED 42 C8 3E 20 02 0B 18 : 7A
3660 F3 C5 3E 10 42 4B EB 21 : 9F
3668 00 00 29 EB 29 EB 30 01 : 59
3670 23 B7 ED 42 30 03 09 18 : 5D
3678 01 13 3D 20 ED EB C1 C9 : D3
---------------------------------
SUM: FC 00 82 C8 23 14 74 A4 1963

3680 2A 2F 30 2E 00 CD 1E 20 : C2
3688 CD 8E 36 C3 10 38 11 CB : 78
3690 31 CD E5 1F CD C4 1F 18 : CA
3698 06 B7 28 F2 CD 33 20 AF : A6
36A0 32 71 30 32 80 30 C3 21 : 99
36A8 20 3A 72 30 B7 CA 21 20 : BE
36B0 E5 D5 C5 3A 71 30 3D 20 : B7
36B8 05 CD DF 36 18 0B 3D 20 : 67
36C0 05 CD CD 36 18 03 CD 44 : 01
36C8 37 C1 D1 E1 C9 2A 78 30 : 45
36D0 7C B5 28 6C 2B 22 78 30 : BA
36D8 ED 5B 29 30 19 18 08 2A : 04
36E0 76 30 23 22 76 30 2B CD : 89
36E8 94 1F B7 28 53 4F FE 20 : 52
36F0 D2 65 37 C3 67 37 CD D0 : 6C
36F8 1F FE 1B 28 1B 2A 7C 30 : 51
---------------------------------
SUM: 0A DE D4 BC DA 78 03 EE C855

3700 7E 4F 23 22 7C 30 B7 28 : 9D
3708 0F FE 45 20 11 7E FE 4E : 4D
3710 20 0C 23 7E FE 44 20 06 : 35
3718 AF 32 72 30 18 26 79 FE : 38
3720 20 38 D3 FE 40 20 2C 2A : DF
3728 7C 30 7E 4F 23 22 7C 30 : 6A
3730 FE 20 38 C2 FE 40 28 1B : 99
3738 3E 1B 32 80 30 79 18 13 : DF
3740 AF 32 71 30 3A 72 30 B7 : 15
3748 20 AC CD B5 37 CD A0 37 : 29
3750 CD 21 20 4F FE 20 38 1A : CD
3758 3A 74 30 B7 20 34 3A 80 : A3
3760 30 FE 1B 28 2D B7 3E 37 : CA
3768 79 32 80 30 F5 CD 79 38 : CE
3770 F1 C9 21 81 30 06 00 09 : 9B
3778 7E 4F FE 1B 20 E9 21 80 : 90
---------------------------------
SUM: 22 E9 00 5E 35 19 50 82 D63B

3780 30 BE 28 E3 32 80 30 3A : 15
3788 74 30 B7 20 DA CD 8D 38 : E7
3790 18 B2 79 FE 40 38 DB FE : 92
3798 60 30 CA D6 40 4F 18 D2 : A9
37A0 3A 2E 30 4F 3A 7E 30 91 : 60
37A8 4F 3A 32 30 81 6F 3A 2D : 42
37B0 30 67 C3 1E 20 2A 32 30 : 24
37B8 25 3A 34 30 D6 0E 85 6F : 9B
37C0 CD 1E 20 2A 7E 30 26 00 : 09
37C8 CD 39 36 11 0C 31 C3 E5 : 32
37D0 1F 2A 32 30 25 3A 34 30 : 6E
37D8 D6 07 85 6F CD 1E 20 2A : 06
37E0 5D 30 CD 39 36 11 0A 31 : 15
37E8 C3 E5 1F 2A 32 30 ED 5B : 9B
37F0 34 30 19 3E F6 85 6F CD : 72
37F8 1E 20 11 56 31 CD E5 1F : A7
---------------------------------
SUM: FB C6 9E 75 48 45 59 56 4E8B

3800 3A 31 30 C6 30 CD F4 1F : 71
3808 CD F1 1F 3E 3E C3 F4 1F : 2F
3810 CD 40 38 CD 62 38 CD 79 : F2
3818 38 2A 2F 30 7D D6 0A 6F : 8D
3820 CD 1E 20 11 17 31 CD E5 : 16
3828 1F 2A 2F 30 7D D6 08 6F : 72
3830 CD 1E 20 CD 06 36 D8 CD : B9
3838 3F 36 11 0A 31 C3 E5 1F : 88
3840 2A 2F 30 2E 00 CD 1E 20 : C2
3848 3E 23 CD F4 1F 3A 31 30 : DC
3850 C6 30 CD F4 1F 3E A5 CD : 86
3858 F4 1F 3A 5C 30 C6 41 C3 : A3
3860 F4 1F 2A 2F 30 2E 04 CD : 9B
3868 1E 20 11 21 31 3A 73 30 : 7E
3870 B7 20 03 11 35 31 C3 E5 : F9
3878 1F 3A 74 30 B7 20 0E 2A : 0C
---------------------------------
SUM: 0E 62 EC 1C D3 62 CE 52 5045

3880 2F 30 2E 17 CD 1E 20 11 : C0
3888 49 31 C3 E5 1F 2A 2F 30 : CA
3890 2E 17 CD 1E 20 3A 74 30 : 2E
3898 26 2A B7 20 02 26 20 7C : EB
38A0 CD F4 1F 11 50 31 C3 E5 : 1A
38A8 1F AF 32 7F 30 32 80 30 : 91
38B0 CD A9 34 ED 5B 10 30 CD : FF
38B8 03 4C CD B0 36 38 1B FE : 53
38C0 20 38 F7 08 3A 73 30 B7 : EB
38C8 C4 4C 3C CD C4 39 08 77 : 95
38D0 CD D2 39 3E 01 32 7F 30 : F8
38D8 18 D9 FE 0D 20 0F AF 32 : 0C
38E0 7E 30 32 2E 30 3C 32 7F : 2B
38E8 30 C9 C3 03 4C FE 1B 20 : 44
38F0 06 12 AF 32 7F 30 C9 CD : 3E
38F8 00 39 C0 CD 09 39 18 B3 : D3
---------------------------------
SUM: 05 AD 95 B7 42 E3 05 7C 690E

3900 21 17 39 01 24 00 ED B1 : 34
3908 C9 87 21 3B 39 85 6F 30 : 09
3910 01 24 7E 23 66 6F E9 01 : 85
3918 02 04 06 07 08 09 0B 0F : 3E
3920 10 11 13 16 1C 1D 20 2A : CD
3928 2B 2C 2E 2F 30 31 32 33 : 7A
3930 34 35 36 37 38 39 3A 3B : BC
3938 3C 3E 3F C3 39 4C 3A A7 : E2
3940 3A BF 3D D2 39 B7 3C 22 : 56
3948 3A B0 3A BB 39 73 3A B7 : 7C
3950 3D C7 3B AF 3E 71 3C 98 : 71
3958 3E 41 3C C1 3B 8B 3A EB : 67
3960 3D 08 3A 8F 3D 6E 3D E0 : D6
3968 3A 2F 3D 78 3C 49 3E FF : E0
3970 3C C3 39 D2 39 08 3A B7 : 3C
3978 3C 78 3C 66 3C FB 3F 34 : 00
---------------------------------
SUM: 76 5F 6E E1 61 B0 F6 56 CA04

3980 40 5A 40 9D 40 E7 3E 62 : 3E
3988 3F 09 3F 9E 3F C3 39 E6 : 46
3990 3B E3 3B DA 3B C3 39 DD : 47
3998 3B E0 3B 34 3B A8 3B AA : 52
39A0 3B AD 3B 73 3B 75 3B 78 : F9
39A8 3B 81 3B 83 3B 86 3B D7 : 4D
39B0 3B D4 3B E9 3B C3 39 EC : 56
39B8 3B EF 3B AF 32 7F 30 32 : 27
39C0 80 30 E1 C9 3A 7E 30 2A : 6C
39C8 10 30 F5 85 6F 30 01 24 : 7E
39D0 F1 C9 3A 7E 30 3C FE FE : DA
39D8 38 02 3E FE 32 7E 30 D0 : 26
39E0 CD FB 39 16 01 7C 90 B9 : DD
39E8 38 08 78 C6 0A 47 16 00 : E5
39F0 18 F3 78 32 2E 30 15 C8 : F0
39F8 C3 A9 34 3A 34 30 4F 3A : C7
---------------------------------
SUM: 7A E1 8C E9 50 DD 33 13 9361

3A00 2E 30 47 3A 7E 30 67 C9 : BD
3A08 3A 7E 30 B7 C8 3D 32 7E : 54
3A10 30 CD FB 39 16 01 7C B8 : 7C
3A18 30 D8 78 D6 0A 47 16 00 : BD
3A20 18 F4 CD C4 39 4F 7E FE : A1
3A28 0D C8 CD F9 35 28 04 23 : 1F
3A30 0C 18 F3 79 32 7E 30 7E : EE
3A38 FE 0D 28 0D CD F9 35 20 : 5B
3A40 04 23 0C 18 F2 79 32 7E : 66
3A48 30 C3 E0 39 CD C4 39 4F : 25
3A50 B7 C8 2B 0D 28 16 7E 2B : 9E
3A58 CD F9 35 20 05 0D 20 F6 : 43
3A60 18 0A 7E 2B CD F9 35 28 : EE
3A68 03 0D 20 F6 79 32 7E 30 : 7F
3A70 C3 11 3A CD 9A 3A 23 0C : DE
3A78 7E FE 0D 20 03 0D 18 04 : D5
---------------------------------
SUM: 0B 01 D0 CF A2 75 09 14 A2B3

3A80 FE 2B 20 F2 79 32 7E 30 : 94
3A88 C3 E0 39 CD 9A 3A 7E 0E : 09
3A90 2B B9 20 02 0E 2D 71 C3 : 75
3A98 A9 34 2A 0E 30 3A 7E 30 : 2D
3AA0 4F 85 6F 30 01 24 C9 3A : 9B
3AA8 7E 30 B7 C8 3D 32 7E 30 : 4A
3AB0 CD C4 39 FE FE 28 22 4F : 5F
3AB8 3E FE 91 4F 06 00 EB 2A : 37
3AC0 78 30 23 1A CD 1D 3B 23 : 2D
3AC8 22 78 30 AF CD 1D 3B 62 : 00
3AD0 6B 23 0C 0D 28 02 ED B0 : 6E
3AD8 2B 36 20 3C 32 7F 30 C9 : 67
3AE0 CD 08 3B C8 3A 7E 30 4F : 0F
3AE8 78 91 D8 3C 47 2A 78 30 : 36
3AF0 23 1A CD 1D 3B 3E 20 12 : D2
3AF8 1B 23 10 F5 22 78 30 AF : BC
---------------------------------
SUM: 20 46 02 3C 65 6A CA 52 471B

3B00 CD 1D 3B 3C 32 7F 30 C9 : 0B
3B08 06 FE 11 FE 00 2A 10 30 : 7D
3B10 19 EB 1A FE 20 C0 1B 10 : 27
3B18 F9 1A FE 20 C9 D5 08 3A : 11
3B20 2B 30 A4 67 B5 20 01 2C : 68
3B28 EB CD CD 3B 19 08 CD 9A : 48
3B30 1F EB D1 C9 2A 2F 30 7D : AA
3B38 2E 00 D6 0C 47 0E 03 ED : 55
3B40 5B 1E 30 C5 D5 E5 CD 1E : 13
3B48 20 EB CD 94 1F 23 FE 20 : CC
3B50 30 03 3E 20 2B CD F4 1F : 9C
3B58 10 F0 E1 D1 C1 14 CD A9 : FD
3B60 36 FE 20 20 03 0D 20 DB : 7F
3B68 CD 1E 20 CD F1 1F 10 FB : F3
3B70 C3 10 38 AF 01 3E 01 01 : FB
3B78 3E 02 CD BB 3B AF C3 9A : 0F
---------------------------------
SUM: 07 32 DD 70 6A A5 E4 EA CF63

3B80 1F AF 01 3E 01 01 3E 02 : 4F
```

▶1599年，長曽我部元親様は無事全国を統一されました。おめでとうございます。うるうる。ついに念願のレベル5統一に成功しました。そのときは感動のあまりしばらくの間，身動きできませんでした。以前は「レベル5を統一するなんて，人間じゃネエ」と言っていた僕ですが，僕はどうやら人間じゃなくなってしまったようです。田口　琢也（15）東京都

```
3B88 CD BB 3B 06 FE CD 94 1F : 47
3B90 B7 28 03 23 10 F7 C0 EB : B7
3B98 CD C4 39 EB 1A CD 9A 1F : 55
3BA0 23 AF CD 9A 1F C3 D2 39 : 26
3BA8 AF 01 3E 01 01 3E 02 CD : FD
3BB0 BB 3B 3E 01 32 71 30 22 : 2A
3BB8 76 30 C9 2A 1E 30 84 67 : D2
3BC0 C9 3E 02 32 71 30 C9 21 : C6
3BC8 00 00 22 78 30 2A 29 30 : 4D
3BD0 AF C3 9A 1F 3E 30 01 3E : D8
3BD8 31 01 3E 32 01 3E 33 01 : 15
3BE0 3E 34 01 3E 35 01 3E 36 : 5B
3BE8 01 3E 37 01 3E 38 01 3E : 2C
3BF0 39 32 40 3C 3E 03 CD CC : C1
3BF8 34 D8 E5 E5 AF ED B1 C1 : E4
-----------------------------------
SUM: C8 EF E3 73 D9 25 97 4B 9684

3C00 B7 ED 42 44 4D E1 D8 C8 : F8
3C08 7E B7 C8 3E 4D ED B1 C0 : E6
3C10 7E FE 43 20 F6 23 0B 78 : 7B
3C18 B1 C8 7E FE 52 20 EC 23 : 76
3C20 0B 78 B1 C8 3A 40 3C BE : 70
3C28 20 E1 3E 0D BE 23 20 FC : 49
3C30 22 7C 30 3E 01 32 72 30 : E1
3C38 AF 3A 74 30 32 80 30 C9 : 38
3C40 34 3A 73 30 EE 01 32 73 : A5
3C48 30 C3 62 38 3A 7E 30 4F : C4
3C50 3E FE B9 C8 91 4F 06 00 : A3
3C58 21 FE 00 19 7E FE 20 C0 : 94
3C60 54 5D 2B ED B8 C9 3A 74 : F8
3C68 30 EE 01 32 74 30 C3 79 : 31
3C70 38 AF 32 7E 30 32 2E 30 : 57
3C78 CD 48 35 DA 80 36 ED 5B : 22
-----------------------------------
SUM: AC B4 7F A3 20 53 1E D0 0CA5

3C80 5F 30 1A B7 C8 CD 76 34 : 9F
3C88 ED 53 5F 30 2A 5D 30 23 : A9
3C90 22 5D 30 ED 5B 61 30 ED : 75
3C98 4B 35 30 06 00 B7 ED 52 : AC
3CA0 B7 ED 42 D8 13 ED 53 61 : 72
3CA8 30 ED 5B 63 30 CD 76 34 : 82
3CB0 ED 53 63 30 C3 AA 33 CD : 40
3CB8 48 35 DA 80 36 2A 5D 30 : C4
3CC0 7C B5 C8 2B 22 5D 30 ED : C0
3CC8 5B 5F 30 CD ED 3C ED 53 : 20
3CD0 5F 30 ED 5B 61 30 B7 ED : 0C
3CD8 52 D0 1B ED 53 61 30 ED : FB
3CE0 5B 63 30 CD ED 3C ED 53 : 24
3CE8 63 30 C3 AA 33 CD 80 34 : B4
3CF0 E5 2A 67 30 B7 ED 52 38 : D4
3CF8 04 ED 5B 67 30 E1 C9 CD : 5A
-----------------------------------
SUM: 04 35 68 13 53 D1 A8 CE E171

3D00 48 35 DA 80 36 ED 5B 63 : B8
3D08 30 1A B7 C8 CD 76 34 ED : 2D
3D10 53 63 30 ED 4B 61 30 03 : B2
3D18 ED 43 61 30 2A 5D 30 B7 : 2F
3D20 ED 42 30 08 ED 43 5D 30 : 24
3D28 ED 53 5F 30 C3 AA 33 CD : 3C
3D30 48 35 DA 80 36 2A 61 30 : C8
3D38 7C B5 C8 2B 22 61 30 ED : C4
3D40 5B 63 30 CD ED 3C ED 53 : 24
3D48 63 30 ED 4B 35 30 06 00 : 36
3D50 09 ED 4B 5D 30 B7 ED 42 : B4
3D58 C2 AA 33 0B ED 43 5D 30 : 67
3D60 ED 5B 5F 30 CD ED 3C ED : BA
3D68 53 5F 30 C3 AA 33 CD 48 : 97
3D70 35 DA 80 36 3A 2E 30 C6 : 23
3D78 10 38 03 32 2E 30 3A 2E : 43
-----------------------------------
SUM: 64 6A 00 23 9E 7D C0 12 2730

3D80 30 47 3A 7E 30 B8 30 04 : 4B
3D88 78 32 7E 30 C3 AA 33 CD : C5
3D90 48 35 DA 80 36 3A 2E 30 : A5
3D98 D6 10 30 01 AF 32 2E 30 : 56
3DA0 47 3A 34 30 80 DA AA 33 : 1C
3DA8 47 3A 7E 30 B8 38 05 05 : 29
3DB0 78 32 7E 30 C3 AA 33 CD : C5
3DB8 48 35 DA 80 36 18 F5 CD : E7
3DC0 48 35 DA 80 36 2A 61 30 : C8
3DC8 ED 5B 63 30 3A 35 30 47 : C1
3DD0 1A B7 28 06 CD 76 34 23 : 99
3DD8 10 F6 22 61 30 ED 53 63 : 5C
3DE0 30 22 5D 30 ED 53 5F 30 : AE
3DE8 C3 AA 33 CD 48 35 DA 80 : 44
3DF0 36 2A 61 30 ED 5B 63 30 : CC
3DF8 7C B5 C8 ED 4B 35 30 06 : 9C
-----------------------------------
SUM: 18 81 0C 70 E3 7C 7A E6 9457

3E00 00 B7 ED 42 30 09 21 00 : 40
3E08 00 ED 5B 67 30 18 06 CD : CA
3E10 ED 3C 0D 20 FA 22 61 30 : 03
3E18 ED 53 63 30 E5 D5 ED 4B : C5
3E20 35 30 06 00 09 ED 5B 5D : 19
3E28 30 B7 ED 52 D1 E1 38 03 : 13
3E30 C2 AA 33 0D 28 09 CD 76 : 20
3E38 34 28 04 23 0D 20 F7 22 : C9
3E40 5D 30 ED 53 5F 30 C3 AA : C9
3E48 33 CD 08 3B F5 2A 78 30 : 0A
3E50 23 3E 0D CD 1D 3B 23 3E : F4
3E58 0E CD 1D 3B 23 F1 28 09 : 78
3E60 04 1A CD 1D 3B 1B 23 10 : 91
3E68 F8 AF CD 1D 3B 22 78 30 : 96
3E70 ED 5B 5F 30 D5 CD 76 34 : 23
3E78 62 6B D1 7E B7 28 10 E5 : F0
-----------------------------------
SUM: 41 83 CB F9 E4 C7 73 BA E98D

3E80 44 4D 2A 65 30 B7 ED 42 : 36
3E88 44 4D E1 03 ED B0 1B AF : DC
3E90 12 ED 53 65 30 C3 AA 33 : 87
3E98 CD 4C 3C CD C4 39 36 0D : 62
3EA0 CD 4D 35 DA 80 36 CD 26 : D2
3EA8 35 CD 4D 35 C3 AA 33 CD : F1
3EB0 4D 35 DA 80 36 2A 5F 30 : CB
3EB8 54 5D CD 76 34 C8 1A B7 : C1
3EC0 C8 1B EB 36 20 06 FE 1A : 42
3EC8 FE 0D 28 06 13 10 F8 36 : 8A
3ED0 0D C9 ED 5B 5F 30 B7 ED : 51
3ED8 52 7D 32 7E 30 AF 32 2E : BE
3EE0 30 CD E0 39 C3 AA 33 CD : 83
3EE8 48 35 DA 80 36 00 00 00 : 0D
3EF0 2A 5D 30 22 7A 30 2A 32 : DF
3EF8 30 25 3A 34 30 D6 08 85 : 56
-----------------------------------
SUM: 01 71 19 C3 23 DA A5 FA 4981

3F00 6F CD 1E 20 3E 2A C3 F4 : 99
3F08 1F CD 48 35 DA 80 36 CD : C6
3F10 16 3F DA C4 1F C9 2A 7A : 7F
3F18 30 23 7C B5 37 C8 2B CD : 7B
3F20 04 35 EB 2A 5D 30 23 CD : CB
3F28 04 35 B7 ED 52 44 4D D8 : 98
3F30 37 C8 C5 D5 2A 1E 30 ED : FE
3F38 5B 1C 30 B7 ED 52 2B B7 : 7F
3F40 ED 42 E1 C1 C5 D5 CD 97 : CF
3F48 1F E1 C1 09 AF CD 9A 1F : FF
3F50 2A 29 30 ED 5B 1C 30 B7 : CE
3F58 ED 52 C0 21 00 00 22 78 : BA
3F60 30 C9 CD 48 35 DA 80 36 : D3
3F68 CD 16 3F DA C4 1F 2A 7A : 83
3F70 30 CD 04 35 E5 2A 5D 30 : D2
3F78 23 CD 04 35 EB 2A 65 30 : D3
-----------------------------------
SUM: E1 61 F9 D5 CC 2A 3E 46 314F

3F80 AF ED 52 44 4D E1 20 03 : 83
3F88 77 18 04 EB 03 ED B0 CD : EB
3F90 EC 34 2A 7A 30 22 5D 30 : A3
3F98 CD 3B 34 C3 AA 33 CD 48 : F1
3FA0 35 DA 80 36 2A 1C 30 CD : 08
3FA8 94 1F B7 C8 01 00 00 CD : 00
3FB0 94 1F B7 28 04 23 03 18 : D4
3FB8 F6 CD 06 36 03 B7 ED 42 : E8
3FC0 DA C4 1F C5 2A 5D 30 CD : 06
3FC8 04 35 C1 7E B7 20 10 ED : 4C
3FD0 5B 1C 30 CD 91 1F CD EC : DD
3FD8 34 CD 3B 34 C3 AA 33 0B : 1B
3FE0 E5 C5 50 59 44 4D 2A 65 : 73
3FE8 30 E5 E5 19 EB E1 B7 ED : 83
3FF0 42 44 4D 03 E1 ED B8 C1 : 1D
3FF8 E1 18 D4 CD 48 35 DA 80 : 71
-----------------------------------
SUM: D7 41 49 4E E9 AF CD 80 DC94

4000 36 CD 2C 33 2A 2F 30 2E : 19
4008 00 CD 1E 20 11 7A 31 CD : 94
4010 E5 1F CD A9 36 D6 30 DA : 90
4018 10 38 FE 08 D2 10 38 21 : 89
4020 FF FF 22 7A 30 32 31 30 : 5D
4028 CD 84 32 CD EC 34 CD D8 : 15
4030 33 C3 8A 33 CD 48 35 DA : D7
4038 80 36 3A 75 30 B7 20 10 : 7C
4040 3C 32 75 30 21 50 1A 22 : C0
4048 34 30 CD CF 32 C3 97 33 : BF
4050 AF 32 75 30 CD 2C 33 C3 : 75
4058 52 33 3A 70 30 B7 C4 48 : 22
4060 35 DA 80 36 2A 32 30 25 : 76
4068 2D CD 1E 20 2A 32 30 CD : 91
4070 C9 40 38 08 22 32 30 CD : 9A
4078 CF 32 18 E8 3A 70 30 B7 : 92
-----------------------------------
SUM: 15 4D 0C D8 5C F0 84 BE 4A30

4080 28 06 CD 1A 33 C3 52 33 : 90
4088 3A 5B 30 32 5A 30 2A 32 : DD
4090 30 22 56 30 2A 34 30 22 : 88
4098 58 30 C3 52 33 3A 70 30 : AA
40A0 B7 C4 48 35 DA 80 36 2A : B2
40A8 32 30 ED 5B 34 30 19 CD : F4
40B0 1E 20 2A 34 30 CD C9 40 : A2
40B8 38 C2 22 34 30 CD CF 32 : 4E
40C0 2A 32 30 ED 4B 34 30 18 : 40
40C8 DE CD A9 36 CD F0 35 FE : 7A
40D0 5A 28 58 FE 58 28 55 FE : AB
40D8 43 28 4E FE 41 28 4F FE : 6D
40E0 53 28 E6 FE 44 28 49 FE : 12
40E8 51 28 4A FE 57 28 47 FE : 85
40F0 45 28 40 FE 1F 28 35 FE : 25
40F8 1D 28 33 FE 1C 28 31 FE : E9
-----------------------------------
SUM: D4 78 B9 DD DF BF 02 2A 1BE5

4100 1E 28 33 FE 31 28 24 FE : F2
4108 32 28 21 FE 33 28 1A FE : EC
4110 34 28 1B FE 35 28 B2 FE : 82
4118 36 28 15 FE 37 28 16 FE : E4
4120 38 28 13 FE 39 28 0C 37 : 15
4128 C9 2C 3E 2D 24 3E 2D 3E : 2D
4130 2C B7 C9 2C 3E 2D 25 B7 : 1F
4138 C9 CD 78 41 ED 7B 0C 30 : F3
4140 CD 7A 42 CD 3B 35 CD F7 : 8A
4148 41 CD DD 41 CD A9 38 CD : A7
4150 D1 35 CD 10 42 3A 80 30 : 0F
4158 FE 0D 20 0C CD 49 43 ED : 7D
4160 5B 10 30 CD 8B 45 18 D4 : 24
4168 CD D6 42 20 D6 CD EC 42 : D6
4170 18 CA DC 99 36 C3 39 41 : CA
4178 2A 56 30 22 32 30 2A 58 : B6
-----------------------------------
SUM: F7 07 A0 62 38 14 9F E4 A4D0

4180 30 22 34 30 CD CF 32 2A : AE
4188 32 30 ED 4B 34 30 CD 00 : CB
4190 4C AF 32 70 30 32 7E 30 : AD
4198 32 2E 30 CD B5 37 CD 7A : 90
41A0 42 CD 10 38 C3 37 42 D5 : 68
41A8 57 1E 00 CB 3A CB 1B 2A : 8A
41B0 26 30 19 D1 C9 3D 3D C5 : 48
41B8 3C 4F 3A 28 30 A1 C1 C9 : 48
41C0 3A 5B 30 CD A7 41 E5 F5 : 54
41C8 CD 94 1F B7 20 0C F1 F5 : 49
41D0 CD B7 41 CD A7 41 AF CD : F6
41D8 9A 1F F1 E1 C9 3A 5A 30 : 18
41E0 67 3A 5B 30 6F 94 30 06 : 65
41E8 3A 28 30 94 85 3C 2A 32 : 43
41F0 30 84 67 22 2C 30 C9 3A : 9C
41F8 5B 30 CD A7 41 ED 5B 10 : 98
-----------------------------------
SUM: 75 74 26 73 74 FD 02 CA 19B9

4200 30 0E 7F CD 94 1F FE 20 : 5B
4208 D8 12 13 23 0D 20 F4 C9 : 0A
4210 CD C0 41 3A 7F 30 B7 20 : 8E
4218 07 CD 94 1F B7 28 13 C9 : 42
4220 ED 5B 10 30 0E 7F 1A CD : FC
4228 9A 1F 13 23 FE 0D C8 0D : CF
4230 20 F4 3E 0D C3 9A 1F CD : A8
4238 A9 34 3A 35 30 47 2A 32 : 1F
4240 30 22 2C 30 3A 5A 30 F5 : 67
4248 CD A7 41 CD 94 1F B7 28 : 14
4250 18 CD FD 41 3E 0D 12 ED : 6D
4258 5B 10 30 CD 03 4C 21 2D : 05
4260 30 34 F1 CD B7 41 10 DF : 09
4268 C9 E1 2A 10 30 36 0D EB : 42
4270 21 2D 30 CD 03 4C 34 10 : DE
4278 FA C9 2A 32 30 25 3A 34 : E2
-----------------------------------
SUM: B0 00 11 C5 FF BE 8C F0 BD47

4280 30 D6 07 85 6F CD 1E 20 : 0C
4288 3A 5B 30 0E 00 CD B5 41 : 96
4290 CD A7 41 08 CD 94 1F B7 : F4
4298 28 04 08 0C 18 EF 69 26 : D6
42A0 00 CD 39 36 11 0A 31 C3 : 4B
42A8 E5 1F E5 F5 CD C4 39 F1 : 99
42B0 77 23 36 0D 21 7E 30 34 : E0
42B8 E1 C9 1A B7 C8 CD AA 42 : FC
42C0 13 18 F7 3A 5B 30 CD A7 : 5B
42C8 41 CD 10 42 CD DD 41 ED : 38
42D0 5B 10 30 CD 03 4C E5 C5 : 61
42D8 21 E3 42 01 09 00 ED B1 : EE
42E0 C1 E1 C9 05 0D 18 1B 1E : CE
42E8 1F 22 23 24 FE 05 28 1F : D2
42F0 FE 0D 28 4E FE 18 28 51 : 10
42F8 FE 1B 28 34 FE 1E 28 0F : C8
-----------------------------------
SUM: 48 B7 A3 8B 56 E2 12 0F C094

4300 FE 1F 28 45 FE 22 CA 83 : F7
4308 43 CD 09 39 C3 78 41 3A : 08
4310 5B 30 47 CD B5 41 CD A7 : 09
4318 41 08 CD 94 1F B7 C8 08 : 50
4320 32 5B 30 08 3A 5A 30 B8 : 41
4328 C0 08 32 5A 30 C3 37 42 : C0
4330 AF 32 7E 30 32 2E 30 3A : 59
4338 5B 30 CD A7 41 3E 0D C3 : 4E
4340 9A 1F AF 32 7E 30 32 2E : A8
4348 30 3A 5B 30 CD B7 41 32 : EC
4350 5B 30 CD C0 41 3A 5A 30 : 1D
4358 CD A7 41 4F CD 94 1F B7 : 3B
4360 20 0A 79 CD B7 41 32 5A : F4
4368 30 C3 37 42 3A 35 30 47 : 52
4370 3A 5B 30 05 28 07 CD B5 : 7B
4378 41 B9 C8 10 F9 32 5A 30 : 87
-----------------------------------
SUM: 96 FA B2 AD DD 7F B9 30 7472

4380 C3 37 42 3A 75 30 B7 28 : FA
4388 0A AF 32 75 30 CD 52 33 : E2
4390 C3 78 41 3C 32 75 30 21 : B0
4398 50 1A 22 34 30 C3 84 41 : 78
43A0 ED 7B 0C 30 3E 0C C3 F4 : A5
43A8 1F 3A 6B 30 EE 01 32 6B : 80
43B0 30 CD 52 33 C3 78 41 1A : 18
43B8 CD F0 35 D6 41 D8 FE 04 : E3
43C0 D0 4F 08 79 CD CC 34 DA : 47
43C8 96 44 08 32 5C 30 21 00 : C1
43D0 00 22 5D 30 CD 2C 33 CD : A8
43D8 84 32 CD EC 34 CD 87 33 : 2A
43E0 C3 78 41 1A 13 FE 2F 20 : F6
43E8 32 1A 13 CD F0 35 D6 41 : 68
43F0 D8 FE 04 D0 87 DD 21 12 : 41
43F8 30 4F 06 00 DD 09 DD 4E : 96
-----------------------------------
SUM: D0 B0 6D 06 C8 A0 03 D5 67D9

4400 02 DD 46 03 13 CD B2 1F : D9
4408 E5 B7 ED 42 E1 38 03 C2 : A9
4410 C4 1F DD 75 00 DD 74 01 : 87
4418 CD 10 34 01 00 03 C5 CD : A7
4420 26 44 C1 0C 10 F8 3A 5C : D5
4428 30 91 3E 20 20 02 3E 2A : A9
4430 CD AA 42 3E 2F CD AA 42 : DF
4438 79 08 79 C6 41 CD AA 42 : BA
4440 3E 3A CD AA 42 79 CD CC : 43
4448 34 38 4B D5 E5 C5 CD 9E : A1
4450 44 1B 3E 2D CD AA 42 EB : 6E
```


▶ウチの学校の先生は，ケンタッキーフライドチキンのことを「からあげ屋」と言って生徒に大笑いされていましたが，服屋の店員が“ハウスマヌカン”と呼ばれるようになり，それにノセられるアルバイターが多い昨今，ウチの先生のような個性は貴重品です。

松本　篤志 (17) 東京都

```
4458 CD 9E 44 3E 3A CD AA 42 : E0
4460 60 69 CD 9E 44 3E 20 CD : A3
4468 AA 42 3E 5B CD AA 42 C1 : FF
4470 E1 AF ED B1 EB E1 60 69 : C3
4478 CD 9E 44 3E 5D CD AA 42 : 03
---------------------------------
SUM: 4F 6D D4 BD 1B C4 AC 89 E2CC

4480 08 CD B0 44 CD 94 1F 23 : 6C
4488 B7 28 05 CD AA 42 18 F4 : A9
4490 CD C3 42 C3 42 43 11 98 : C3
4498 31 CD BA 42 18 F2 D5 EB : C4
44A0 CD C4 39 EB CD BC 44 21 : A3
44A8 7E 30 34 34 34 34 D1 C9 : 18
44B0 3C 11 20 00 2A 20 30 19 : 00
44B8 3D 20 FC C9 7C CD C1 44 : 70
44C0 7D F5 0F 0F 0F 0F CD CA : 45
44C8 44 F1 E6 0F F6 30 FE 3A : 88
44D0 38 02 C6 07 12 13 C9 11 : 06
44D8 6C 30 3A 5C 30 83 5F 30 : 74
44E0 01 14 2A 67 30 7E B7 C8 : D3
44E8 12 36 00 3A 5C 30 CD B0 : 8B
44F0 44 CD 9A 1F CD 52 33 C3 : DF
44F8 78 41 3A 5C 30 11 6C 30 : 2C
---------------------------------
SUM: B5 1A 2D 9B 48 CE 39 91 EA72

4500 83 5F 30 01 14 2A 67 30 : E8
4508 7E B7 C0 1A B7 C8 77 CD : D2
4510 EC 34 CD 52 33 C3 78 41 : EE
4518 0E 00 06 04 CD 23 45 0E : 5B
4520 04 06 04 CD 2F 45 0C 10 : 6B
4528 FA CD C3 42 C3 42 43 3A : 4E
4530 31 30 91 3E 20 20 02 3E : B0
4538 2A CD AA 42 79 C6 30 CD : 1F
4540 AA 42 3E 2F CD AA 42 79 : 8B
4548 87 81 2A 20 30 85 6F 30 : A6
4550 01 24 CD 94 1F C6 41 CD : 79
4558 AA 42 3E 3A C3 AA 42 D6 : E9
4560 30 FE 08 D0 D5 F5 CD 2C : C9
4568 33 F1 32 31 30 CD 84 32 : 3A
4570 CD EC 34 CD 8A 33 CD D8 : 1C
4578 33 CD 78 41 D1 18 0C 21 : CF
---------------------------------
SUM: 93 EB 1E 2C 95 F1 7A 44 2F0D

4580 00 00 22 5D 30 2A 67 30 : 70
4588 22 5F 30 D5 3A 5B 30 CD : 18
4590 B5 41 CD A7 41 CD FD 41 : B6
4598 3E 0D 12 AF 13 12 D1 1A : 1C
45A0 13 CD F0 35 CD 32 36 30 : 6A
45A8 B6 FE 42 28 D2 FE 21 CA : D9
45B0 A0 43 FE 25 CA A9 43 FE : BA
45B8 26 CA D7 44 FE 2F CA B7 : B9
45C0 43 FE 52 CA FA 44 FE 4D : E6
45C8 CA E3 43 FE 4E CA 18 45 : 63
45D0 FE 45 CA FE 45 FE 4C CA : 64
45D8 6D 46 FE 53 CA 48 47 FE : 5B
45E0 46 CA 06 48 FE 43 CA EB : 54
45E8 48 FE 44 CA 1D 4A FE 58 : 11
45F0 CA 6F 4A FE 50 CA 46 4B : 2C
45F8 00 00 00 00 00 C9 D5 CD : 6B
---------------------------------
SUM: 74 28 29 77 E7 E0 55 BC 96FC

4600 84 32 CD EC 34 D1 2A 5D : FB
4608 30 CD E9 35 CD 32 36 D4 : 24
4610 12 36 22 5D 30 CD 3B 34 : 33
4618 CD 97 33 21 FF FF 22 7A : 52
4620 30 3E 01 32 70 30 CD 26 : 34
4628 35 CD 59 46 CD 29 38 CD : 9C
4630 D1 37 CD A9 38 2A 10 30 : 20
4638 7E FE 1B 28 08 3A 80 30 : B1
4640 CD 09 39 18 E1 CD 2C 33 : 34
4648 3A 75 30 B7 20 05 CD AA : 32
4650 33 18 03 CD 52 33 C3 78 : DB
4658 41 ED 4B 61 30 2A 5D 30 : C1
4660 B7 ED 42 7D 2A 32 30 84 : 73
4668 67 22 2C 30 C9 CD B5 46 : 76
4670 38 33 20 3B 3E 0C CD F4 : D1
4678 1F CD 84 32 CD EC 34 CD : 5C
---------------------------------
SUM: 37 9E 16 FF 2E B2 51 42 E85A

4680 06 36 ED 4B 72 1F AF ED : A1
4688 42 38 1A 11 A1 31 CD E5 : 29
4690 1F CD 9D 1F 2A 65 30 22 : 89
4698 70 1F CD A6 1F 38 06 CD : 2C
46A0 AF 46 C3 00 47 F5 3E 0C : 3E
46A8 CD F4 1F F1 CD 99 36 CD : 3A
46B0 52 33 C3 78 41 CD EC 34 : EE
46B8 CD E9 35 3E 04 CD A3 1F : BC
46C0 CD 09 20 C8 FE 08 37 C0 : BB
46C8 CD D7 46 D8 C0 CD 09 20 : 78
46D0 C8 FE 08 28 F3 37 C9 3E : 27
46D8 0C CD F4 1F CD 06 20 D8 : B7
46E0 11 5E 31 CD E5 1F ED 5B : B9
46E8 76 1F CD D3 1F 1A FE 1B : 87
46F0 20 02 B7 C9 13 13 13 13 : EE
46F8 13 3E 04 CD A3 1F AF C9 : 5C
---------------------------------
SUM: 9A 18 66 E5 ED 92 8B 35 D41D

4700 3A 5C 30 CD B0 44 3E 20 : E5
4708 CD 40 47 3E 2A CD AA 42 : 75
4710 3A 5D 1F CD 40 47 3E 3A : 82
4718 CD 40 47 ED 5B 74 1F 06 : 35
4720 0D CD 34 47 3E 2E CD 40 : CE
4728 47 06 03 CD 34 47 CD C3 : 28
4730 42 C3 42 43 13 1A CD AA : 2E
4738 42 CD 9A 1F 23 10 F5 C9 : B9
4740 CD AA 42 CD 9A 1F 23 C9 : 2B
4748 CD EC 34 CD E9 35 FE 0D : E3
4750 28 62 D5 3E 04 CD A3 1F : 30
4758 D1 13 1A FE 3A 20 0F 13 : 78
4760 1A FE 20 30 09 CD D7 46 : 5B
4768 DA A5 46 C2 AF 46 3E 0C : C6
4770 CD F4 1F 11 BE 31 CD E5 : 92
4778 1F CD 9D 1F CD 84 32 CD : F8
---------------------------------
SUM: 59 0B 77 33 21 74 88 24 BACD

4780 EC 34 ED 5B 67 30 2A 65 : 8E
4788 30 23 AF ED 52 DA A5 46 : 06
4790 22 72 1F 21 00 00 22 70 : 66
4798 1F 22 6E 1F CD AF 1F DA : 43
47A0 A5 46 2A 67 30 22 70 1F : 5D
47A8 CD AC 1F DA A5 46 CD AF : D9
47B0 46 C3 00 47 3A 5C 30 CD : E3
47B8 B0 44 CD 94 1F B7 28 A5 : F8
47C0 3E 20 CD AA 42 CD 94 1F : 97
47C8 23 B7 28 05 CD AA 42 18 : D8
47D0 F4 3E 3A CD AA 42 11 AE : E4
47D8 31 CD BA 42 CD E0 39 CD : AD
47E0 C3 42 CD A0 37 CD A9 36 : 55
47E8 FE 1B CA 42 43 FE 0D 28 : 9B
47F0 06 CD 42 43 C3 65 47 ED : B4
47F8 5B 10 30 3E 04 CD A3 1F : 6C
---------------------------------
SUM: 6D 00 31 C5 7B CA 65 51 FC9F

4800 CD 42 43 C3 6E 47 CD E9 : 80
4808 35 CD 32 36 38 06 CD 12 : 87
4810 36 22 5D 30 CD E9 35 2A : FA
4818 22 30 CD C5 48 D8 CD 3B : 0C
4820 34 2A 5D 30 E5 CD 84 32 : 53
4828 CD EC 34 E1 22 5D 30 CD : 4A
4830 97 33 CD 83 48 20 2C AF : 5D
4838 32 2E 30 CD E0 39 CD B5 : F8
4840 37 CD D1 37 CD 3B 34 CD : 15
4848 AA 33 CD 59 46 CD A0 37 : ED
4850 CD A9 36 FE 1B CA 78 41 : 48
4858 FE 20 CA 21 46 21 7E 30 : 1E
4860 34 18 CF AF 32 7E 30 32 : DC
4868 2E 30 2A 5D 30 23 22 5D : B7
4870 30 ED 5B 5F 30 CD 76 34 : 7E
4878 ED 53 5F 30 1A B7 20 B2 : 72
---------------------------------
SUM: 4F 29 7E 99 0A A9 FB AD A875

4880 C3 78 41 2A 5F 30 3A 7E : ED
4888 30 4F 85 6F 30 01 24 EB : B3
4890 2A 22 30 CD 94 1F EB F5 : DC
4898 BE 20 0C CD B3 48 20 07 : D9
48A0 79 32 7E 30 F1 AF C9 7E : 40
48A8 23 0C FE 20 38 03 F1 18 : 91
48B0 E6 D1 C9 EB 2A 22 30 D5 : BC
48B8 CD 94 1F EB BE EB 13 23 : 4A
48C0 28 F6 E1 B7 C9 1A 13 FE : AA
48C8 22 20 18 06 7F 1A FE 0D : 04
48D0 28 0B FE 22 28 07 CD 9A : E9
48D8 1F 13 23 10 F0 AF CD 9A : 6B
48E0 1F B7 C9 CD 94 1F B7 37 : 0D
48E8 C8 B7 C9 2A 22 30 AF 47 : BA
48F0 32 1C 4A CD 9A 1F 10 FB : 29
48F8 1A FE 2F 20 04 32 1C 4A : 03
---------------------------------
SUM: EE 68 8B 2C 9B E1 A3 F5 7B09

4900 13 CD E9 35 CD 32 36 38 : 6B
4908 06 CD 12 36 22 5D 30 CD : 97
4910 E9 35 2A 22 30 CD C5 48 : 74
4918 D8 2A 24 30 CD C5 48 D8 : 08
4920 CD 3B 34 3A 1C 4A B7 28 : BB
4928 11 2A 5D 30 E5 CD 84 32 : 30
4930 CD EC 34 E1 22 5D 30 CD : 4A
4938 97 33 CD 83 48 20 41 3A : FD
4940 1C 4A B7 28 27 AF 32 2E : 7B
4948 30 CD E0 39 CD B5 37 CD : 9C
4950 D1 37 CD 3B 34 CD AA 33 : EE
4958 CD 59 46 CD A0 37 CD A9 : 86
4960 36 FE 1B 28 38 FE 20 20 : ED
4968 11 CD C4 1F CD 26 35 CD : B6
4970 A3 49 30 06 CD 80 36 C3 : 68
4978 9D 49 21 7E 30 34 18 BA : BB
---------------------------------
SUM: 8D 81 B5 BF 21 F5 A2 C7 3704

4980 AF 32 7E 30 32 2E 30 2A : 49
4988 5D 30 23 22 5D 30 ED 5B : A7
4990 5F 30 CD 76 34 ED 53 5F : A5
4998 30 1A B7 20 9D CD 87 33 : 45
49A0 C3 78 41 2A 22 30 CD 0D : D2
49A8 4A EB 2A 24 30 CD 0D 4A : D7
49B0 44 4D B7 ED 52 28 43 38 : 2A
49B8 1F EB 2A 10 30 01 FE 00 : 73
49C0 09 E5 E5 B7 ED 52 EB CD : 81
49C8 C4 39 44 4D E1 B7 ED 42 : 55
49D0 44 4D EB D1 ED B8 18 22 : 2C
49D8 E5 CD C4 39 EB 19 EB 09 : A7
49E0 E5 2A 10 30 01 FE 00 09 : 57
49E8 B7 ED 52 44 4D 03 E1 EB : 56
49F0 ED B0 C1 EB 36 20 23 0C : CE
49F8 20 FA CD C4 39 EB 2A 24 : 1D
---------------------------------
SUM: AA 40 39 64 97 24 1B 04 B40E

4A00 30 CD 94 1F B7 CA 4D 35 : B3
4A08 12 13 23 18 F4 0E 00 CD : 2F
4A10 94 1F B7 20 03 69 67 C9 : 26
4A18 23 0C 18 F3 00 CD E9 35 : 25
4A20 CD 32 36 D8 CD 12 36 E5 : 07
4A28 CD E9 35 CD 32 36 30 02 : 52
4A30 E1 C9 CD 12 36 23 D1 EB : 9E
4A38 22 7A 30 CD 04 35 EB CD : 8A
4A40 04 35 E5 B7 ED 52 E1 C8 : BD
4A48 D8 7E B7 20 03 12 18 10 : 6A
4A50 E5 44 4D 2A 65 30 B7 ED : D9
4A58 42 44 4D E1 D8 03 ED B0 : 2C
4A60 2A 7A 30 22 5D 30 CD 2C : 7C
4A68 33 CD 87 33 C3 78 41 1A : 50
4A70 13 CD F0 35 D6 41 DA C4 : BA
4A78 1F FE 04 D2 C4 1F 32 3C : 44
---------------------------------
SUM: 28 B6 CF 0C CE 4D 76 5A 314B

4A80 4B 4F CD E9 35 CD F0 35 : 77
4A88 D6 41 DA C4 1F FE 04 D2 : A8
4A90 C4 1F B9 CA C4 1F 32 3D : B8
4A98 4B CD CC 34 DA C4 1F 22 : F7
4AA0 40 4B ED 43 44 4B 3A 3C : C0
4AA8 4B CD CC 34 DA C4 1F 22 : F7
4AB0 3E 4B ED 43 42 4B 2A 44 : B4
4AB8 4B AF ED 42 38 01 3C 2A : C8
4AC0 3E 4B B7 28 09 2A 40 4B : 26
4AC8 ED 4B 44 4B 50 59 B7 20 : 47
4AD0 06 ED 5B 44 4B 18 04 ED : E6
4AD8 5B 42 4B E5 AF ED B1 C1 : DB
4AE0 C2 C4 1F B7 ED 42 EB B7 : 2D
4AE8 ED 52 DA C4 1F 2A 42 4B : B3
4AF0 54 5D ED 4B 44 4B B7 ED : 1C
4AF8 42 30 02 42 4B 2A 3E 4B : B4
---------------------------------
SUM: 15 F6 48 4B 78 72 D2 85 E1C4

4B00 ED 5B 40 4B 7E 08 1A 77 : EA
4B08 08 12 23 13 0B 78 B1 20 : A4
4B10 F3 3A 3C 4B CD B0 44 E5 : 5A
4B18 3A 3D 4B CD B0 44 D1 06 : 5A
4B20 20 CD 94 1F 08 EB CD 94 : F4
4B28 1F 08 CD 9A 1F EB 08 CD : 6D
4B30 9A 1F 13 23 10 EB CD 52 : 09
4B38 33 C3 78 41 00 00 00 00 : AF
4B40 00 00 00 00 00 00 CD FC : C9
4B48 33 3A D9 30 B7 28 06 AF : 0A
4B50 11 E8 31 18 04 3C 11 E0 : 73
4B58 31 32 D9 30 CD BA 42 CD : 02
4B60 C3 42 C3 42 43 00 00 00 : 4D
4B68 00 00 00 00 00 00 00 00 : 00
4B70 00 00 00 00 00 00 00 00 : 00
4B78 00 00 00 00 00 00 00 00 : 00
---------------------------------
SUM: 66 31 7C 4D 08 53 A8 8D 0603

4B80 00 00 00 00 00 00 00 00 : 00
4B88 00 00 00 00 00 00 00 00 : 00
4B90 00 00 00 00 00 00 00 00 : 00
4B98 00 00 00 00 00 00 00 00 : 00
4BA0 00 00 00 00 00 00 00 00 : 00
4BA8 00 00 00 00 00 00 00 00 : 00
4BB0 00 00 00 00 00 00 00 00 : 00
4BB8 00 00 00 00 00 00 00 00 : 00
4BC0 00 00 00 00 00 00 00 00 : 00
4BC8 00 00 00 00 00 00 00 00 : 00
4BD0 00 00 00 00 00 00 00 00 : 00
4BD8 00 00 00 00 00 00 00 00 : 00
4BE0 00 00 00 00 00 00 00 00 : 00
4BE8 00 00 00 00 00 00 00 00 : 00
4BF0 00 00 00 00 00 00 00 00 : 00
4BF8 00 00 00 00 00 00 00 00 : 00
---------------------------------
SUM: 00 00 00 00 00 00 00 00 0000

4C00 C3 06 4C C3 62 4C E5 C5 : 30
4C08 25 2D CD 1E 20 3E 2B CD : 93
4C10 F4 1F 3E 2D CD 59 4C 3E : 2E
4C18 2B CD F4 1F E5 79 D6 0F : 4E
4C20 85 6F CD 1E 20 11 10 31 : 51
4C28 CD E5 1F E1 24 CD 1E 20 : E1
4C30 3E 6C CD F4 1F 3E 1C CD : B1
4C38 59 4C 3E 6C CD F4 1F 10 : 3F
4C40 EB 24 CD 1E 20 3E 2B CD : 50
4C48 F4 1F 3E 2D CD 59 4C 3E : 2E
4C50 2B CD F4 1F C1 E1 C3 1E : 8E
4C58 20 C5 CD F4 1F 0D 20 FA : EC
4C60 C1 C9 C5 D5 E5 F5 2A 2C : 54
4C68 30 CD 1E 20 3A 34 30 47 : 20
4C70 3A 2E 30 4F B7 28 09 1A : E9
4C78 FE 0D 28 11 13 0D 20 F7 : 7B
---------------------------------
SUM: 43 D1 49 3F 1A 4F 78 B4 63B2

4C80 1A FE 0D 28 08 CD F4 1F : 35
4C88 13 10 F5 18 05 CD F1 1F : 12
4C90 10 FB F1 E1 D1 C1 C9 00 : 38
4C98 00 00 00 00 00 00 00 00 : 00
4CA0 00 00 00 00 00 00 00 00 : 00
4CA8 00 00 00 00 00 00 00 00 : 00
4CB0 00 00 00 00 00 00 00 00 : 00
4CB8 00 00 00 00 00 00 00 00 : 00
4CC0 00 00 00 00 00 00 00 00 : 00
4CC8 00 00 00 00 00 00 00 00 : 00
4CD0 00 00 00 00 00 00 00 00 : 00
4CD8 00 00 00 00 00 00 00 00 : 00
4CE0 00 00 00 00 00 00 00 00 : 00
4CE8 00 00 00 00 00 00 00 00 : 00
4CF0 00 00 00 00 00 00 00 00 : 00
4CF8 00 00 00 00 00 00 00 00 : 00
---------------------------------
SUM: 3D 09 F3 21 DE 5B AE 3E FBC8
```

▶チクショウ,「冒険！　コロンブス号」に行くことができなかった。せっかく招待状をもらったのに，なんで部活なんかあるの。部長はつれぇなあ。でも「2001年カプセル」に，Oh！X宛てのメッセージを出していたら，絶対2001年にはウケていただろうに……。

田中　哲也（17）兵庫県

リスト2 WINERソースリスト

```
0000                            1 ;-------------------------------
0000                            2 ;     WINER  ver 1.0   page 0
0000                            3 ;         < screen editer >
0000                            4 ;     Copyright in 1988 by 環
0000                            5 ;-------------------------------
0000                            6
0000                            7 program EQU 03000H
0000                            8 makadr  EQU 0CA00H
0000                            9
0000                           10 ;         --------------------
0000                           11                 OFFSET  makadr-program
0000                           12 ;         ORG      program
0000                           13 ;         DS       2000H
0000                           14 ;         --------------------
0000                           15
0000                           16 ;/*/* メインメモリー */*/
0000                           17 macnsub  EQU 04C00H  ;機種別サブルーチン
0000                           18 _tabadr  EQU 04E00H  ;タブフォーマット格納アドレス
0000                           19 _linebf  EQU 04F00H  ;ライン入力バッファアドレス
0000                           20 _txtadrA EQU 05000H  ;テキストA格納アドレス
0000                           21 _txtadrB EQU 0A000H  ;テキストB格納アドレス
0000                           22 _txtadrC EQU 0D000H  ;テキストC格納アドレス
0000                           23 _txtadrD EQU 0E500H  ;テキストD格納アドレス
0000                           24
0000                           25 ;/*/* 特種ワークエリア */*/
0000                           26 _cutbf   EQU 00000H  ;カット&ペースト用バッファアドレス
0000                           27 _copybf  EQU 00700H  ;コピーバッファアドレス
0000                           28 _txtwbf  EQU 00A00H  ;ウィンドウ用のテキストワーク
0000                           29                              ;  退避アドレス
0000                           30 _seachbf EQU 00B00H  ;サーチ用バッファアドレス
0000                           31 _chengbf EQU 00B80H  ;チェンジ用バッファアドレス
0000                           32 _combf   EQU 00C00H  ;コマンドレベル用スクリーンワーク
0000                           33 _commsk  EQU 007H    ;
0000                           34 _delbf   EQU 00000H  ;デリートバッファアドレス
0000                           35 _delmsk  EQU 003H
0000                           36
0000                           37 ;*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*/*
0000                           38
0000                           39 ;         --------------
3000                           40                 ORG     program
3000                           41 ;         --------------
3000                           42
3000 00 00 00 00 00 00 00      43                 DS 12   ;       JP COLD
3007 00 00 00 00 00
300C                           44                         ;       JP HOT1
300C                           45                         ;       JP HOT2
300C                           46                         ;       拡張用
300C                           47 ;------ work area ------
300C 00 00                     48 stack   DW 0
300E                           49 ;*** main memory map ***
300E 00 4E                     50 tabadr  DW _tabadr
3010 00 4F                     51 linebf  DW _linebf
3012                           52             ;
3012 00 50                     53 txtst   DW _txtadrA ; テキストエリアAスタートアドレス
3014 00 A0                     54             DW _txtadrB ; テキストエリアBスタートアドレス
3016                           55                         ;   /テキストエリアAエンドアドレス
3016 00 D0                     56             DW _txtadrC ; テキストエリアCスタートアドレス
3018                           57                         ;   /テキストエリアBエンドアドレス
3018 00 E5                     58             DW _txtadrD ; テキストエリアDスタートアドレス
301A                           59                         ;   /テキストエリアCエンドアドレス
301A FF FF                     60             DW  0FFFFH  ;   /テキストエリアDエンドアドレス
301C                           61
301C                           62 ;*** EX work map ***
301C 00 00                     63 cutbf   DW _cutbf   ; カット&ペーストバッファ アドレス
301E 00 07                     64 copybf  DW _copybf  ; コピー用バッファ アドレス
3020 00 0A                     65 txtwbf  DW _txtwbf  ; テキストワークバッファアドレス
3022 00 0B                     66 seachbf DW _seachbf ; サーチ用文字列バッファアドレス
3024 80 0B                     67 chengbf DW _chengbf ; チェンジ用文字列バッファアドレス
3026 00 0C                     68 combf   DW _combf   ; コマンドテキストバッファアドレス
3028 07                        69 commsk  DB _commsk  ; コマンドテキストラインマスク
3029 00 00                     70 delbf   DW _delbf   ; デリートバッファアドレス
302B 03                        71 delmsk  DB _delmsk  ; デリートバッファアドレスマスク
302C                           72 ;*** line print work ***
302C 00                        73 l_curx  DB 0    ; ラインプリントX座標
302D 00                        74 l_cury  DB 0    ; ラインプリントY座標
302E 00                        75 l_offs  DB 0    ; 1行表示の オフセット
302F                           76
302F                           77 ;*** window work ***
302F 00 00                     78 scrnsiz DW 0    ; スクリーンサイズ
3031 00                        79 windno  DB 0    ; ウィンドウNo.
3032                           80
3032 00                        81 xstart  DB 0    ; ウィンドウ  X座標スタート
3033 00                        82 ystart  DB 0    ; ウィンドウ  Y座標スタート
3034 00                        83 xsize   DB 0    ; ウィンドウ  X座標サイズ
3035 00                        84 ysize   DB 0    ; ウィンドウ  Y座標サイズ
3036                           85
3036 01 02 4E 14 01 02 4E      86 windata DB 01,02,78,20:01,02,78,20  ; ウィンドウデータ
303D 14
303E 01 02 4E 14 01 02 4E      87             DB 01,02,78,20:01,02,78,20  ; 退避用ワーク  8ch
3045 14
3046 01 02 4E 14 01 02 4E      88             DB 01,02,78,20:01,02,78,20
304D 14
304E 01 02 4E 14 01 02 4E      89             DB 01,02,78,20:01,02,78,20
3055 14
3056                           90
3056                           91 ;*** command level work ***
3056 18 07 2E 06               92 comwind DB 24,07,46,06              ; コマンド入力用ウィンドウデータ
305A 00                        93 comtop  DB  0                       ; コマンウィンドウのトップライン
305B 00                        94 comlin  DB  0                       ; コマンドラインNo
305C                           95
305C                           96 ;*** text work ***
305C 00                        97 nowtxt  DB 0    ; 現在のエディットテキスト  0=A 1=B 2=C 3=D
305D 00 00                     98 lineno  DW 0    ; 現在の行No.
305F                           99             ;
305F 00 00                    100 lineadr DW 0    ; 現在の行アドレス
3061 00 00                    101 dtopno  DW 0    ; ディスプレイのトップNo.
3063 00 00                    102 dtopadr DW 0    ; ディスプレイのトップアドレス
3065 00 00                    103 endadr  DW 0    ; 最終行アドレス
3067 00 00                    104 ntxtst  DW 0    ; 現在のテキストエリアのスタートアドレス
3069 00 00                    105 ntxtend DW 0    ; 現在のテキストエリアのエンドアドレス
306B 00                       106 tabsw   DB 0    ; タブデータ表示スイッチ
306C                          107
306C 3B 3B 3B 3B              108 txtchr  DM ";;;;"
3070                          109 ;*** mode work ***
3070 00                       110 getlmode DB 0   ; ライン入力  モード
3071                          111                     ; 0=command level
3071                          112                     ; 1=screen edit
3071 00                       113 inkymode DB 0   ; 1文字入力  モード
3072                          114                     ; 0=normal
3072                          115                     ; 1=put out from copy buf
3072                          116                     ; 2=put out from delite buf
3072 00                       117 mcroflog DB 0   ; 0=normal
3073                          118                     ; 1=key macro
3073 00                       119 insmode  DB 0   ; インサート/オーバーライト  モードフラグ
3074 00                       120 ctrlkeep DB 0   ; コントロールキー  キープフラグ
3075 00                       121 windmxf  DB 0   ; ウィンドウMAXフラグ
3076                          122 ;*** copy & delite buff work ***
3076 00 00                    123 copypnt DW 0    ; コピー出力ポインター
3078 00 00                    124 delpnt  DW 0    ; デリートバッファ  スタックポインター
307A 00 00                    125 markbf  DW 0    ; カット&ペーストマークバッファ
307C 00 00                    126 macrpnt DW 0    ; キー入力マクロポインター
307E                          127 ;*** key work ***
307E 00                       128 pointer DB 0    ; カーソル  ポインター
307F 00                       129 inpchk  DB 0    ; キー入力  チェック  フラグ
3080 00                       130 chrbf   DB 0    ; キャラクターバッファ
3081                          131 keyconf0        ; キー  コンフィギュレーション
3081 00 01 02 03 04 05 06     132         DB 00H,01H,02H,03H,04H,05H,06H,07H ;00H-1FH @ A B C D E F G
3088 07
3089 08 09 0A 0B 0C 0D 0E     133         DB 08H,09H,0AH,0BH,0CH,0DH,0EH,0FH ;        H I J K L M N O
3090 0F
3091 10 11 12 13 14 15 16     134         DB 10H,11H,12H,13H,14H,15H,16H,17H ;        P Q R S T U V W
3098 17
3099 18 19 1A 1B 1C 1D 1E     135         DB 18H,19H,1AH,1BH,1CH,1DH,1EH,1FH ;        X Y Z
30A0 1F
30A1 20 21 22 23 24 25 26     136         DB 20H,21H,22H,23H,24H,25H,26H,27H ;20H-3FH   ! " # $ % & '
30A8 27
30A9 28 29 2A 2B 2C 2D 2E     137         DB 28H,29H,2AH,2BH,2CH,2DH,2EH,2FH ;        ( ) * + , - . /
30B0 2F
30B1 30 31 32 33 34 35 36     138         DB 30H,31H,32H,33H,34H,35H,36H,37H ;        0 1 2 3 4 5 6 7
30B8 37
30B9 38 39 3A 3B 3C 3D 3E     139         DB 38H,39H,3AH,3BH,3CH,3DH,3EH,3FH ;        8 9 : ; < = > ?
30C0 3F
30C1                          140 ;*** for ctrl ***
30C1 20 3A 3B 2C 3D 2B 2D     141 skipchr DM " :;,=+-*/"      ;skiped chractor
30C8 2A 2F
30CA 3C 3E 28 29 5B 5D 7B     142             DM "<>()[]{}「」"    ; all is 24
30D1 7D A2 A3
30D4 22 27 20 20 20           143             DB 22H  DM "'   "
30D9                          144
30D9                          145 ;**** Window prw buff ****
30D9 01                       146 wprwsw  DB 1
30DA FF FF FF FF              147 winprw  DB 0FFH,0FFH,0FFH,0FFH
30DE FF FF FF FF FF           148             DB 0FFH,0FFH,0FFH,0FFH,0FFH
30E3                          149 ;**** ASCII data ****
30E3 3C 3C 20 57 20 49 20     150 titlem  DM "<< W I N E R >> ver 1.0 (C) 1988 Tama."  DB 0
30EA 4E 20 45 20 52 20 3E
30F1 3E 20 76 65 72 20 31
30F8 2E 30 20 28 43 29 20
30FF 31 39 38 38 20 54 61
3106 6D 61 2E 00
310A 00 00 00 00 00 00        151 namare  DS 6
3110 5B 20 20 20 20 20 5D     152 data1m  DM "[     ]"
3117 5B 20 20 20 20 20 20     153 data2m  DM "[       ]" DB 0
311E 20 5D 00
3121 20 2F 20 49 6E 73 65     154 insertm DM " / Insert mode   / " DB 0
3128 72 74 20 6D 6F 64 65
312F 20 20 20 2F 20 00
3135 20 2F 20 4F 76 65 72     155 overtpm DM " / Overtype mode / " DB 0
313C 74 79 70 65 20 6D 6F
3143 64 65 20 2F 20 00
3149 4E 6F 72 6D 61 6C 00     156 narmalm DM "Normal" DB 0
3150 43 74 72 6C 20 00        157 ctrlm   DM "Ctrl "  DB 0
3156 3C 20 57 69 6E 64 2E     158 windm   DM "< Wind." DB 0
315D 00
315E 48 69 74 20 3C 43 52     159 dirm    DM "Hit <CR> on File or Break" DB 1EH,0DH,0
3165 3E 20 6F 6E 20 46 69
316C 6C 65 20 6F 72 20 42
3173 72 65 61 6B 1E 0D 00
317A 57 68 69 63 68 20 77     160 wjumpm  DM "Which window do you jump ?   " DB 0
3181 69 6E 64 6F 77 20 64
3188 6F 20 79 6F 75 20 6A
318F 75 6D 70 20 3F 20 20
3196 20 00
3198 4E 6F 20 75 73 69 6E     161 nousem  DM "No using" DB 0
319F 67 00
31A1 4E 6F 77 20 6C 6F 61     162 loadm   DM "Now loading " DB 0
31A8 64 69 6E 67 20 00
31AE 20 44 6F 20 79 6F 75     163 save?   DM " Do you save ? " DB 0
31B5 20 73 61 76 65 20 3F
31BC 20 00
31BE 4E 6F 77 20 73 61 76     164 savem   DM "Now saving  " DB 0
31C5 69 6E 67 20 20 00
31CB 4D 65 6D 6F 72 79 20     165 memovr  DM "Memory Over          " DB 0
31D2 4F 76 65 72 20 20 20
31D9 20 20 20 20 20 20 00
31E0 20 50 72 77 20 6F 6E     166 prwonm  DM " Prw on"  DB 0
31E7 00
31E8 20 50 72 77 20 6F 66     167 prwofm  DM " Prw off" DB 0
31EF 66 00
31F1                          168
31F1                          169 PAGE1   ;for page 1 LABEL
31F1                          170
31F1                          171
31F1                          172
31F1                          173
31F1                          174
31F1                          175 ;----------------------
31F1                          176
31F1                          177 ;         -------------
4C00                          178                 ORG    macnsub
4C00                          179 ;         -------------
4C00                          180
4C00 C3 06 4C                 181 OWINDOW JP openwindow
4C03 C3 62 4C                 182 LNPRINT JP lineprint
4C06                          183
4C06                          184 openwindow ;in H=locate Y  L=locate X
4C06                          185                 ;   B=Y size    C=X size
4C06                          186                 ;des AF,HL,BC,DE
4C06                          187
4C06 E5 C5                    188             PUSH HL:PUSH BC
4C08 25 2D CD 1E 20           189              DEC H:DEC L:CALL #LOC
4C0D 3E 2B CD F4 1F           190              LD A,"+":CALL #PRINT
4C12 3E 2D CD 59 4C           191              LD A,"-":CALL opnwin2
4C17 3E 2B CD F4 1F           192              LD A,"+":CALL #PRINT
4C1C E5                       193              PUSH HL
4C1D 79 D6 0F                 194               LD A,C:SUB 15
4C20 85 6F CD 1E 20           195               ADD A,L:LD L,A:CALL #LOC
4C25 11 10 31 CD E5 1F        196               LD DE,data1m:CALL #MSX
4C2B E1                       197              POP HL
4C2C                          198              ;
4C2C 24 CD 1E 20              199 opnwin1  INC H:CALL #LOC
4C30 3E 6C CD F4 1F           200              LD A,"l":CALL #PRINT
4C35 3E 1C CD 59 4C           201              LD A,1CH:CALL opnwin2
4C3A 3E 6C CD F4 1F           202              LD A,"l":CALL #PRINT
4C3F 10 EB                    203              DJNZ opnwin1
4C41 24 CD 1E 20              204              INC H:CALL #LOC
4C45                          205              ;
4C45 3E 2B CD F4 1F           206              LD A,"+":CALL #PRINT
4C4A 3E 2D CD 59 4C           207              LD A,"-":CALL opnwin2
4C4F 3E 2B CD F4 1F           208              LD A,"+":CALL #PRINT
4C54 C1 E1 C3 1E 20           209             POP BC:POP HL:JP #LOC
4C59 C5 CD F4 1F              210 opnwin2 PUSH BC:CALL #PRINT
4C5D 0D 20 FA                 211              IF DEC(C)<>0 JR opnwin2+1
4C60 C1 C9                    212             POP BC:RET
4C62                          213
4C62                          214 lineprint ; ラインプリント
4C62                          215                 ; in DE=ASCII adress /des noting
4C62                          216
4C62 C5 D5                    217            PUSH BC:PUSH DE
4C64 E5 F5                    218            PUSH HL:PUSH AF
4C66 2A 2C 30 CD 1E 20        219             LD HL,(l_curx):CALL #LOC
4C6C 3A 34 30 47              220             LD A,(xsize):LD B,A
4C70 3A 2E 30 4F              221             LD A,(l_offs):LD C,A
4C74 B7 28 09                 222             IF A=0 JR l_prt1
4C77 1A FE 0D 28 11           223 l_prt0  LD A,(DE):IF A=0DH JR l_prt2
4C7C 13 0D 20 F7              224             INC DE:IF DEC(C)<>0 JR l_prt0
4C80 1A FE 0D 28 08           225 l_prt1  LD A,(DE):IF A=0DH JR l_prt2
4C85 CD F4 1F 13              226             CALL #PRINT:INC DE
4C89 10 F5 18 05              227             DJNZ l_prt1:JR l_prt3
4C8D CD F1 1F 10 FB           228 l_prt2  CALL #PRNTS:DJNZ l_prt2
4C92 F1 E1                    229 l_prt3 POP AF:POP HL
4C94 D1 C1 C9                 230            POP DE:POP BC:RET
```

THE SENTINEL

▶ STUDIO Xにひと言。雪印のアイスは"パモナ"だ。決してパナモではない。さて，ウケている理由というのは中村由真のCMだからなのだ。　　小宮　崇（16）埼玉県

```
0000                    1 ;------------------------------
0000                    2 ;    WINER            page 1
0000                    3 ;       < screen editer >
0000                    4 ;    Copyright in 1988 by 環
0000                    5 ;------------------------------
0000                    6
0000                    7 ;         --------------------
0000                    8               OFFSET  makadr-program
3000                    9               ORG     program
3000                   10 ;         --------------------
3000                   11
3000 CD F1 31          12               CALL COLD
3003 CD 0C 32          13               CALL HOT#1
3006 C3 50 32          14               JP   HOT#2
3009                   15
3009                   16 ;         ------------
31F1                   17               ORG     PAGE1
31F1                   18 ;         ------------
31F1                   19
31F1                   20
31F1                   21
31F1                   22 COLD   ;コールドスタート
31F1 CD 10 34          23               CALL TAREAI
31F4 06 04 11 6F 30    24               LD B,4:LD DE,txtchr+3
31F9 78 3D             25 COLD1   LD A,B:DEC A
31FB D5 C5             26               PUSH DE:PUSH BC
31FD CD CC 34          27                CALL TEXTDAT
3200 C1 D1 38 04       28               POP BC:POP DE:JR C,COLD2
3204 7E 36 00 12       29               LD A,(HL):LD (HL),0:LD (DE),A
3208 1B 10 EE          30 COLD2   DEC DE:DJNZ COLD1
320B C9                31               RET
320C                   32 ;
320C                   33 HOT#1   ;ホットスタート1
320C 2A 20 30 AF 47    34               LD HL,(txtwbf):XOR A:LD B,A     ;text work clear !
3211 CD 9A 1F 23 10 FA 35               CALL #POKE:INC HL:DJNZ HOT#1+5
3217                   36               ;
3217 AF 32 31 30       37 HOT#11  XOR A:LD (windno),A              ;op No.init
321B                   38               ;
321B 21 00 00 CD 9A 1F 39               LD HL,0:CALL #POKE              ;cut & peast buff  clear
 !
3221                   40               ;
3221 2A 1E 30          41               LD HL,(copybf)
3224 CD 9A 1F 24       42               CALL #POKE:INC H                ;copy buf clear !
3228 CD 9A 1F 24 CD 9A 1F      43       CALL #POKE:INC H:CALL #POKE
322F                   44               ;
322F 21 00 00 22 78 30 45               LD HL,0:LD (delpnt),HL          ;delite buf clear !
3235                   46               ;
3235 CD CC 34          47               CALL TEXTDAT                    ;Text aria over check
3238 22 67 30 ED 53 69 30      48       LD (ntxtst),HL:LD (ntxtend),DE
323F CD EC 34 D0       49               CALL ENDCHK:RET NC
3243 2A 1A 30 22 18 30 50               LD HL,(txtst+8):LD (txtst+6),HL
3249 22 16 30 22 14 30 51               LD (txtst+4),HL:LD (txtst+2),HL
324F C9                52               RET
3250                   53
3250                   54 HOT#2   ;ホットスタート2
3250 ED 73 0C 30       55               LD (stack),SP
3254                   56               ;
3254 AF 67 6F          57               XOR A:LD H,A:LD L,A             ;Data etc. init
3257 32 5B 30 32 5A 30 58               LD (comlin),A:LD (comtop),A
325D 22 70 30 22 73 30 59               LD (getlmode),HL:LD (insmode),HL
3263 22 7F 30          60               LD (inpchk),HL
3266                   61
3266 2A 26 30 11 80 00 62               LD HL,(combf):LD DE,080H        ;command window clear!
326C 3A 28 30 47       63               LD A,(commsk):LD B,A
3270 04 AF             64               INC B:XOR A
3272 CD 9A 1F 19 10 FA 65 HOT#21  CALL #POKE:ADD HL,DE:DJNZ HOT#21
3278                   66
3278 CD FC 33          67               CALL PRWINIT                    ;window prw init
327B                   68
327B CD 44 33 CD 52 33 69               CALL SCRNSIZ:CALL SCRNINIT_c
3281                   70               ;
3281 C3                71               DB 0C3H
3282 00 00             72 TO_COM  DW 0
3284                   73
3284                   74 ;-----------------------
3284                   75 WINDPOP ;ウィンドウデータセット &
3284                   76               ; in  (windno) /  ウィンドウサイズチェック
3284                   77               ; out nothing
3284                   78               ; des AF,HL,DE,BC
3284                   79
3284 AF 32 75 30       80               XOR A:LD (windmxf),A    ; window position
3288 3A 31 30          81               LD A,(windno)           ;    & size set
328B FE 08 38 01 AF    82               IF A>=8 THEN XOR A
3290 32 31 30          83               LD (windno),A
3293 87 87 21 36 30    84               ADD A,A:ADD A,A:LD HL,windata
3298 85 6F             85               ADD A,L:LD L,A
329A 11 32 30 01 04 00 ED      86       LD DE,xstart:LD BC,4:LDIR
32A1 B0
32A2 3A 31 30          87               LD A,(windno)
32A5                   88
32A5 6F 87 85          89               LD L,A:ADD A,A:ADD A,L  ; text No.
32A8 2A 20 30 85       90               LD HL,(txtwbf):ADD A,L  ;   & line No. set
32AC 6F 30 01 24       91               LD L,A:IF C THEN INC H
32B0 EB 21 5C 30       92               EX DE,HL:LD HL,nowtxt
32B4 01 03 00 CD 91 1F 93               LD BC,3:CALL #PEEK@
32BA                   94
32BA 3A 5C 30          95               LD A,(nowtxt)           ; if unusing text
32BD CD CC 34          96               CALL TEXTDAT            ;  then command level
32C0 30 06 CD C4 1F C3 81      97       IF C THEN CALL #BELL:JP TO_COM-1
32C7 32
32C8 22 67 30          98               LD (ntxtst),HL          ; text area data set
32CB ED 53 69 30       99               LD (ntxtend),DE         ;
32CF                  100
32CF                  101 ;********
32CF                  102 WINDCHK ;ウィンドウサイズチェック
32CF                  103               ; in  (xsize),(ysize),(xstart),(ystart)
32CF                  104               ; out (xsize),(ysize),(xstart),(ystart)
32CF                  105               ; des AF,HL,DE,BC
32CF                  106
32CF 2A 34 30 ED 5B 32 30     107       LD HL,(xsize):LD DE,(xstart)
32D6 ED 4B 2F 30      108 WINDCHK0 LD BC,(scrnsiz)
32DA 7D FE 10 30 02 2E 10     109       LD A,L:IF A<16 THEN LD L,16
32E1 24 25 20 01 24   110               IF H=0 THEN INC H
32E6 1C 1D 20 01 1C   111               IF E=0 THEN INC E
32EB 14 15 20 01 14   112               IF D=0 THEN INC D
32F0 7B 85            113 WINDCHK1 LD A,E:ADD A,L
32F2 B9 38 04         114               IF A<C JR WINDCHK2
32F5 1D 20 F8         115               DEC E:JR NZ,WINDCHK1
32F8 1C               116               INC E
32F9 7B 85            117 WINDCHK2 LD A,E:ADD A,L
32FB B9 38 03 2D 18 F8        118       IF A>=C THEN DEC L:JR WINDCHK2
3301 7A 84            119 WINDCHK3 LD A,D:ADD A,H
3303 B8 38 04         120               IF A<B JR WINDCHK4
3306 15 20 F8         121               DEC D:JR NZ,WINDCHK3
3309 14               122               INC D
330A 7A 84            123 WINDCHK4 LD A,D:ADD A,H
330C B8 38 03 25 18 F8        124       IF A>=B THEN DEC H:JR WINDCHK4
3312 22 34 30 ED 53 32 30     125       LD (xsize),HL:LD  (xstart),DE
3319 C9               126               RET
331A                  127 ;********
331A                  128 WINDPUSH :ウィンドウデータ退避
331A                  129                ;in  (windno)
331A                  130                ;out notihng
331A                  131                ;des AF,HL,DE,BC
331A                  132
331A 3A 31 30         133               LD A,(windno)
331D 87 87 11 36 30   134               ADD A,A:ADD A,A:LD DE,windata
3322 83 5F            135               ADD A,E:LD E,A
3324 21 32 30 01 04 00 ED     136       LD HL,xstart:LD BC,4:LDIR
332B B0
332C                  137 ;********
332C                  138 TXTPUSH :テキストデータ  退避
332C                  139               ; in  (windno)
332C                  140               ; out notihng
332C                  141               ; des AF,HL,DE,BC
332C                  142
332C 3A 31 30         143               LD A,(windno)
332F 6F 87 85         144               LD L,A:ADD A,A:ADD A,L
3332 2A 20 30 85      145               LD HL,(txtwbf):ADD A,L
3336 6F 30 01 24      146               LD L,A:IF C THEN INC H
333A 11 5C 30 01 03 00        147       LD DE,nowtxt:LD BC,3
3340 EB C3 97 1F      148               EX DE,HL:JP   #POKE@
3344                  149 ;********
3344                  150 SCRNSIZ : スクリーンサイズ  セット
3344                  151               ; in  nothing
3344                  152               ; out (scrnsiz)
3344                  153               ; des AF
3344                  154
3344 3A 5C 1F 32 2F 30        155       LD A,(#WIDTH):LD (scrnsiz),A
334A 3A 5B 1F 3D 32 30 30     156       LD A,(#MAXLIN):DEC A:LD (scrnsiz+1),A
3351 C9               157               RET
3352                  158 ;********
3352                  159 SCRNINIT_c
3352                  160               ;画面を消してスクリーン初期化
3352                  161               ; des AF,HL,DE,BC
3352                  162
3352 3E 0C CD F4 1F   163               LD A,0CH:CALL #PRINT    ;cls
3357 3A 5C 1F 21 18 00        164       LD A,(#WIDTH):LD HL,24
335D FE 50 28 02 2E 01        165       IF A<>80 THEN LD L,1    ;title print
3363 CD 1E 20 11 E3 30 CD     166       CALL #LOC:LD DE,titlem:CALL #MSX
336A E5 1F
336C 3A D9 30 B7 28 15        167       LD A,(wprwsw):IF A=0 JR SCRNINIT
3372 CD D8 33 11 DA 30        168       CALL PRWSET:LD DE,winprw
3378                  169 SCRNINIT_c1                        ;window display
3378 1A 3C C8         170               LD A,(DE):INC A:RET Z   ;    by prw
337B D5               171               PUSH DE
337C 3D 32 31 30      172                DEC A:LD (windno),A
3380 CD 87 33         173                CALL SCRNINIT
3383 D1 13            174               POP DE:INC DE
3385 18 F1            175               JR   SCRNINIT_c1
3387                  176 ;********
3387                  177 SCRNINIT; スクリーン初期化       ;a window dsiplay
3387                  178               ; des AF,HL,DE,BC
3387                  179
3387 CD 84 32 CD 3B 34        180       CALL WINDPOP:CALL LINSET
338D CD 10 36 AF      181               CALL PDATA:XOR A
3391 32 2E 30 32 7E 30        182       LD (l_offs),A:LD (pointer),A
3397                  183 ;********
3397                  184 WTXTPRT ; ウィンドウ表示してテキスト表示
3397                  185               ; des AF,HL,DE,BC
3397                  186
3397 2A 32 30 ED 4B 34 30     187       LD HL,(xstart):LD BC,(xsize)
339E CD 00 4C CD EB 37        188       CALL OWINDOW:CALL PWINDNO
33A4 CD B5 37 CD D1 37        189       CALL PCURDAT:CALL PLINDAT
33AA                  190 ;********
33AA                  191 TXTPRT  ; テキスト表示
33AA                  192               ; des AF,HL,DE,BC
33AA                  193
33AA CD A9 34         194               CALL TABPRT
33AD 2A 32 30 22 2C 30        195       LD HL,(xstart):LD (l_curx),HL
33B3 3A 35 30 47      196               LD A,(ysize):LD B,A
33B7 ED 5B 63 30 21 2D 30     197       LD DE,(dtopadr):LD HL,l_cury
33BE 1A B7 28 0A      198 TXTPRT1 LD A,(DE):IF A=0 JR TXTPRT2
33C2 CD 03 4C CD 76 34        199       CALL LNPRINT:CALL NXTLIN
33C8 34 10 F3         200               INC (HL):DJNZ TXTPRT1
33CB C9               201               RET
33CC 3E 0D 12         202 TXTPRT2 LD A,0DH:LD (DE),A
33CF CD 03 4C 34      203 TXTPRT3 CALL LNPRINT:INC (HL)
33D3 10 FA            204               DJNZ TXTPRT3
33D5 AF 12            205               XOR A:LD (DE),A
33D7 C9               206               RET
33D8                  207 ;********
33D8                  208 PRWSET  ; ウィンドウ プライオリティ セット
33D8                  209               ; in  (windno)
33D8                  210               ; out (winprw)
33D8                  211               ; des AF,HL,BC,E
33D8                  212
33D8 3A 31 30 5F      213               LD A,(windno):LD E,A
33DC 01 FF 08 21 DA 30        214       LD BC,008FFH:LD HL,winprw
33E2 7B               215 PRWSET1 LD A,E
33E3 BE 28 0A         216               IF A=(HL) JR PRWSET2
33E6 79               217               LD A,C
33E7 BE 28 0C         218               IF A=(HL) JR PRWSET3
33EA 23 10 F5         219               INC HL:DJNZ PRWSET1
33ED C3 81 32         220               JP TO_COM-1
33F0 23 7E 2B         221 PRWSET2 INC HL:LD A,(HL):DEC HL
33F3 B9 20 02         222               IF A<>C JR PRWSET4
33F6 73 C9            223 PRWSET3 LD (HL),E:RET
33F8 77 23            224 PRWSET4 LD (HL),A:INC HL
33FA 18 F4            225               JR PRWSET2
33FC                  226 ;********
33FC                  227 PRWINIT ; ウィンドウ プライオリティ 初期化
33FC                  228               ; in  nothing
33FC                  229               ; out nothing
33FC                  230               ; des AF,HL,DE,BC
33FC                  231
33FC 21 DB 30 54 5D 13        232       LD HL,winprw+1:LD DE,HL:INC DE
3402 01 07 00 36 FF ED B0     233       LD BC,7:LD (HL),0FFH:LDIR
3409 3A 31 30 32 DA 30        234       LD A,(windno):LD (winprw),A
340F C9               235               RET
3410                  236 ;********
3410                  237 TAREAI  ; テキストエリア初期化
3410                  238               ; (#MEMAX)にあわせて テキストエリアを調整する
3410                  239               ; in  nothing
3410                  240               ; out text area data work
3410                  241               ; des HL,DE,BC,IX
3410                  242
3410 2A 6A 1F 22 1A 30        243       LD HL,(#MEMAX):LD (txtst+8),HL
3416 06 04 DD 21 18 30        244       LD B,4:LD IX,txtst+6
341C DD 5E 00 DD 56 01        245 TAREAI1 LD E,(IX+0):LD D,(IX+1)
3422 DD 6E 02 DD 66 03        246       LD L,(IX+2):LD H,(IX+3) ;Text clear !
3428 B7 ED 52 30 07   247               SUB HL,DE:JR NC,TAREAI2 ; & PUSH Text top chr
342D 19 DD 75 00 DD 74 01     248       ADD HL,DE:LD (IX+0),L:LD (IX+1),H
3434 DD 2B DD 2B 10 E2        249 TAREAI2 DEC IX:DEC IX:DJNZ TAREAI1
343A C9               250               RET
343B                  251 ;********
343B                  252 LINSET  ; 行No.にあわせてテキスト表示データをセット
343B                  253               ; in  (lineno)
343B                  254               ; out (lineadr),(dtopno),(dtopadr)
343B                  255               ; des AF,HL,DE,BC
343B                  256
343B 2A 5D 30 CD 04 35        257       LD HL,(lineno):CALL LINADR
3441 22 5F 30 ED 43 5D 30     258       LD (lineadr),HL:LD (lineno),BC
3448 EB 60 69         259               EX DE,HL:LD HL,BC
344B 3A 35 30 CB 3F 4F 47     260       LD A,(ysize):SRL A:LD C,A:LD B,A
3452 04 18 03         261               INC B:JR LINSET2+3
3455 CD 80 34 10 FB   262 LINSET2 CALL DNXTLIN:DJNZ LINSET2
345A B7 ED 42 30 03 21 00     263       SUB HL,BC:IF C THEN LD HL,0
3461 00
3462 22 61 30 2A 67 30        264       LD (dtopno),HL:LD HL,(ntxtst)
3468 B7 ED 52 38 04 ED 5B     265       SUB HL,DE:IF NC THEN LD DE,(ntxtst)
346F 67 30
3471 ED 53 63 30 C9   266               LD (dtopadr),DE:RET
3476                  267 ;********
3476                  268 NXTLIN  ; 1行先にスキップ
3476                  269               ; in  DE=ASCII adress
3476                  270               ; out DE=ASCII adress
3476                  271               ; des nothing / flog Z=END
3476                  272
3476 1A B7 C8         273               LD A,(DE):IF A=0 RET
3479 13 FE 0D 20 F8   274               INC DE:IF A<>0DH JR NXTLIN
```

▶源平をやれば「情けなやー」と言われ続け，グラディウスでは未だに3面を見たことのないこの私でも，R-TYPEとイシターは欲しいのだ。そうそう，アフターバーナーも欲しいと思っているので，発売前に3ページくらい使って，ぜひ特集してください。

内藤　陽一（21）愛知県

```
347E B7 C9               275             OR A:RET
3480                     276 ;********
3480                     277 DNXTLIN ; 1行後ろにバック
3480                     278             ; in  DE=ASCII adress
3480                     279             ; out DE=ASCII adress
3480                     280             ; des nothing / flog NC=TOP
3480                     281
3480 1B                  282             DEC DE
3481 1B 1A               283             DEC DE:LD A,(DE)
3483 B7 28 04            284             IF A=0    JR TABINIT-2
3486 FE 0D 20 F7         285             IF A<>0DH JR DNXTLIN+1
348A 13 C9               286             INC DE:RET
348C                     287 ;********
348C                     288 TABINIT ; タブバッファ初期化
348C                     289             ; des AF,HL,BC
348C                     290
348C 2A 0E 30 7E         291             LD HL,(tabadr):LD A,(HL)
3490 FE 2D C8            292             IF A="-" RET
3493 FE 2B C8            293             IF A="+" RET
3496 0E 20               294             LD C,32
3498 36 2B 23            295 TABINIT0 LD (HL),"+":INC HL
349B 06 07 36 2D         296 TABINIT1 LD B,7:LD (HL),"-"
349F 23 10 FB            297             INC HL:DJNZ TABINIT1+2
34A2 0D 20 F3            298             IF DEC(C)<>0 JR TABINIT0
34A5 2B 36 0D C9         299             DEC HL:LD (HL),0DH:RET
34A9                     300 ;
34A9                     301 TABPRT  ; タブデータ表示
34A9                     302             ; in  window data
34A9                     303             ; des AF,HL,DE,BC
34A9                     304
34A9 CD 8C 34            305             CALL TABINIT
34AC 3A 6B 30 B7 C8      306             LD A,(tabsw):IF A=0 RET
34B1 2A 2C 30 E5         307             LD HL,(l_curx):PUSH HL
34B5 2A 32 30 3A 35 30   308              LD HL,(xstart):LD A,(ysize)
34BB 84 67 22 2C 30      309              ADD A,H:LD H,A:LD (l_curx),HL
34C0 ED 5B 0E 30 CD 03 4C 310             LD DE,(tabadr):CALL LNPRINT
34C7 E1 22 2C 30         311             POP HL:LD (l_curx),HL
34CB C9                  312             RET
34CC                     313 ;********
34CC                     314 TEXTDAT ; テキストエリア  データ
34CC                     315             ; in  Acc=text No.
34CC                     316             ; out HL=text start adress
34CC                     317             ;     DE=text end adress
34CC                     318             ;     BC=text size
34CC                     319             ; flog C=unuse / des AF,HL,DE,BC
34CC                     320
34CC FE 04 3F D8         321             CP 4:CCF:RET C
34D0 21 12 30 87         322             LD HL,txtst:ADD A,A
34D4 85 6F               323             ADD A,L:LD L,A
34D6 5E 23               324             LD E,(HL):INC HL
34D8 56 23               325             LD D,(HL):INC HL
34DA 7E 23               326             LD A,(HL):INC HL
34DC 66 6F B7 ED 52      327             LD H,(HL):LD L,A:SUB HL,DE
34E1 F5                  328             PUSH AF
34E2 44 4D 19            329              LD BC,HL:ADD HL,DE
34E5 F1                  330             POP AF
34E6 EB D8               331             EX DE,HL:RET C
34E8 37 C8               332             SCF:RET Z
34EA B7 C9               333             OR A:RET
34EC                     334 ;********
34EC                     335 ENDCHK  ; テキストエリアのエンドチェック
34EC                     336             ; in (ntxtst),(ntxtend)
34EC                     337             ; des AF,HL,BC
34EC                     338             ; flog C=text area over
34EC                     339
34EC ED 4B 67 30         340             LD BC,(ntxtst)
34F0 C5                  341             PUSH BC
34F1 2A 69 30            342              LD HL,(ntxtend)
34F4 B7 ED 42 44 4D      343              SUB HL,BC:LD BC,HL
34F9 E1 AF ED B1         344             POP HL:XOR A:CPIR
34FD 2B 22 65 30 C8      345             DEC HL:LD (endadr),HL:RET Z
3502 37 C9               346             SCF:RET
3504                     347 ;********
3504                     348 LINADR  ; ラインNoの  アドレスを返す
3504                     349             ; in HL=lineno
3504                     350             ; out HL=line adress BC=lineno
3504                     351             ; des AF
3504                     352
3504 D5 EB               353             PUSH DE:EX DE,HL
3506 2A 67 30 01 00 00   354              LD HL,(ntxtst):LD BC,0
350C 7A B3 28 0F         355              IF DE=0 JR LINADR2
3510 7E B7 28 0D         356 LINADR1  LD A,(HL):IF A=0 JR LINADR2+2
3514 23 FE 0D 20 F7      357              INC HL:IF A<>0DH JR LINADR1
3519 03 1B 7A B3 20 F1   358              INC BC:IF DEC(DE)<>0 JR LINADR1
351F D1 C9               359 LINADR2 POP DE:RET
3521 22 65 30 18 F9      360              LD (endadr),HL:JR LINADR2
3526                     361 ;********
3526                     362 LINTRS  ; ラインバッファにテキストを送る
3526                     363             ; in (lineadr),(linebf)
3526                     364             ; des AF,HL,DE,B
3526                     365
3526 ED 5B 5F 30 2A 10 30 366             LD DE,(lineadr):LD HL,(linebf)
352D 06 FF               367             LD B,255
352F 1A                  368 LINTRS1 LD A,(DE)
3530 FE 20 38 0C         369             IF A<20H JR LBFCLA1
3534 77 13 23            370             LD (HL),A:INC DE:INC HL
3537 10 F6 18 0A         371             DJNZ LINTRS1:JR LBFCLA2
353B                     372 ;********
353B                     373 LBFCLA  ; ラインバッファの初期化
353B                     374             ; in (linebf)
353B                     375             ; des HL,B
353B                     376
353B 2A 10 30 06 FF      377             LD HL,(linebf):LD B,255
3540 36 20 23            378 LBFCLA1 LD (HL)," ":INC HL
3543 10 FB               379             DJNZ LBFCLA1
3545 36 0D C9            380 LBFCLA2 LD (HL),0DH:RET
3548                     381 ;********
3548                     382 WRTLIN  ; テキスト書き込み
3548                     383             ; in (lineadr),(endadr)
3548                     384             ; des AF,HL,DE.BC,HL',DE',BC'
3548                     385             ; C=ERROR
3548                     386
3548 3A 7F 30 B7 C8      387             LD A,(inpchk):IF A=0 RET   ;input check
354D CD D1 35 60 69      388 WRTLIN0 CALL SET0DH:LD HL,BC
3552 E5                  389             PUSH HL
3553 ED 5B 5F 30 1A      390              LD DE,(lineadr):LD A,(DE)
3558 B7 20 0F            391              IF A<>0 JR WRTLIN1
355B C1                  392             POP BC
355C CD BB 35 D8         393             CALL WRTCHK:RET C
3560 CD 73 35            394             CALL WRTLIN2
3563 AF 12               395             XOR A:LD (DE),A          ;new line write end adress
3565 ED 53 65 30 C9      396             LD (endadr),DE:RET
356A                     397             ;
356A CD C6 35            398 WRTLIN1  CALL LINSIZ
356D B7 ED 42 20 0C      399              SUB HL,BC:JR NZ,WRTLIN3   ;new size = old size
3572 C1                  400             POP BC
3573 2A 10 30 ED 5B 5F 30 401 WRTLIN2 LD HL,(linebf):LD DE,(lineadr)
357A ED B0 B7 C9         402             LDIR:OR A:RET
357E                     403             ;
357E 30 18               404 WRTLIN3  JR NC,WRTLIN4
3580 19 EB               405              ADD HL,DE:EX DE,HL        ;new size < old size
3582 E5                  406              PUSH HL
3583 44 4D 2A 65 30      407               LD BC,HL:LD HL,(endadr)
3588 B7 ED 42 44 4D 03   408               SUB HL,BC:LD BC,HL:INC BC
358E E1 ED B0            409              POP HL:LDIR
3591 1B ED 53 65 30      410              DEC DE:LD (endadr),DE
3596 18 DA               411              JR WRTLIN2-1
3598                     412              ;
3598 44 4D CD BB 35      413 WRTLIN4  LD BC,HL:CALL WRTCHK
359D 30 02 C1 C9         414              IF C THEN POP BC:RET
35A1 ED 4B 65 30 09      415              LD BC,(endadr):ADD HL,BC  ;new size > old size
35A6 E5                  416              PUSH HL
35A7 60 69 B7 ED 52      417               LD HL,BC:SUB HL,DE
35AC 44 4D 03            418               LD BC,HL:INC BC
35AF D1                  419              POP DE
35B0 2A 65 30 ED 53 65 30 420             LD HL,(endadr):LD (endadr),DE
35B7 ED B8 18 B7         421              LDDR:JR WRTLIN2-1
35BB                     422              ;
35BB E5 D5               423 WRTCHK  PUSH HL:PUSH DE
35BD CD 06 36 AF ED 42   424              CALL LEFBYT:XOR A:SBC HL,BC
35C3 D1 E1 C9            425             POP DE:POP HL:RET
35C6                     426 ;********
35C6                     427 LINSIZ  ; ラインサイズをBCに返す
35C6                     428             ; in  DE=line adress /des AF,DE
35C6                     429             ; out BC=line size
35C6                     430
35C6 01 00 00            431             LD BC,0
35C9 1A 03 13            432             LD A,(DE):INC BC:INC DE
35CC FE 0D 20 F9         433             IF A<>0DH JR LINSIZ+3
35D0 C9                  434             RET
35D1                     435 ;********
35D1                     436 SET0DH  ; ラインバッファの文字列の最後に 0DHを書きこむ
35D1                     437             ; out BC=line size
35D1                     438             ; des AF,HL,BC
35D1                     439
35D1 01 00 01 2A 10 30   440             LD BC,100H:LD HL,(linebf)
35D7 09 2B               441             ADD HL,BC:DEC HL
35D9 2B 7E FE 20 20 06   442 SET0DH1 DEC HL:LD A,(HL):IF A<>" " JR SET0DH2
35DF 0B 78 B1 20 F5      443             IF DEC(BC)<>0 JR SET0DH1
35E4 03                  444             INC BC
35E5 23 36 0D C9         445 SET0DH2 INC HL:LD (HL),0DH:RET
35E9                     446 ;********
35E9                     447 SPSKIP  ; スペース  スキップ
35E9                     448             ; in DE=ASCII adress /des AF
35E9                     449
35E9 1A FE 20 C0         450             LD A,(DE):IF A<>" " RET
35ED 13 18 F9            451             INC DE:JR SPSKIP
35F0                     452 ;********
35F0                     453 UPPER   ; 小文字を大文字に変換
35F0 FE 61 D8            454             IF A<"a" RET
35F3 FE 7B D0            455             IF A>="z"+1 RET
35F6 D6 20 C9            456             SUB 20H:RET
35F9                     457 ;********
35F9                     458 ?SKIP   ; スキップするキャラクターか?
35F9                     459             ; in Acc=check chr.
35F9                     460
35F9 E5 C5               461             PUSH HL:PUSH BC
35FB 21 C1 30 01 18 00 ED 462             LD HL,skipchr:LD BC,24:CPIR
3602 B1
3603 C1 E1 C9            463             POP BC:POP HL:RET
3606                     464 ;********
3606                     465 LEFBYT  ; 残りのバイト数
3606                     466             ; out HL=left byte /des AF,DE
3606                     467             ; flog C=error
3606                     468
3606 2A 69 30 ED 5B 65 30 469             LD HL,(ntxtend):LD DE,(endadr)
360D 13 B7 ED 52 C9      470             INC DE:SUB HL,DE:RET
3612                     471 ;********
3612                     472 GETNUM  ; 10進アスキー→16進  変換
3612                     473             ; in DE=ASCII adr /out HL=hh
3612                     474             ; des AF,DE,BC
3612                     475
3612 CD 1B 36            476             CALL GETNUM1
3615 7C B5 28 01 2B      477             IF HL<>0 THEN DEC HL
361A C9                  478             RET
361B 21 00 00 1A         479 GETNUM1 LD HL,0:LD A,(DE)
361F CD 32 36 D8         480             CALL CHKNUM:RET C
3623 29 44 4D            481             ADD HL,HL:LD BC,HL
3626 29 29 09            482             ADD HL,HL:ADD HL,HL:ADD HL,BC
3629 D6 30 4F 06 00      483             SUB "0":LD C,A:LD B,0
362E 09 13 18 EC         484             ADD HL,BC:INC DE:JR GETNUM1+3
3632                     485 ;********
3632                     486 CHKNUM  ;10進アスキー チェック
3632                     487             ; in Acc
3632                     488             ; out flog C=Error
3632                     489
3632 FE 30 D8            490             CP "0":RET C
3635 FE 3A 3F C9         491             CP "9"+1:CCF:RET
3639                     492 ;********
3639                     493 SEISU   ; 16進→10進アスキー  変換
3639                     494             ; in HL=hh /out DE=ASCII adr
3639                     495             ; des AF,HL,BC
3639                     496
3639 23 7C B5 20 01 2B   497             IF INC(HL)=0 THEN DEC HL
363F                     498
363F 01 0F 31 AF         499 SEISU0  LD BC,namare+5:XOR A
3643 02 0B               500             LD (BC),A:DEC BC
3645 11 0A 00 CD 61 36   501 SEISU1  LD DE,10:CALL SEISU3
364B 7B C6 30            502             LD A,E:ADD A,"0"
364E 02 0B               503             LD (BC),A:DEC BC
3650 7D B4 20 F1         504             LD A,L:OR H:JR NZ,SEISU1
3654 21 09 31            505 SEISU2  LD HL,namare-1
3657 B7 ED 42 C8         506             SUB HL,BC:RET Z
365B 3E 20 02            507             LD A,20H:LD (BC),A
365E 0B 18 F3            508             DEC BC:JR SEISU2
3661 C5                  509 SEISU3  PUSH BC
3662 3E 10 42 4B         510              LD A,16:LD BC,DE
3666 EB 21 00 00         511              EX DE,HL:LD HL,0
366A 29 EB               512 SEISU4   ADD HL,HL:EX DE,HL
366C 29 EB               513              ADD HL,HL:EX DE,HL
366E 30 01 23            514              IF C THEN INC HL
3671 B7 ED 42 30 03      515              SUB HL,BC:JR NC,SEISU5
3676 09 18 01            516              ADD HL,BC:JR SEISU5+1
3679 13 3D 20 ED         517 SEISU5   INC DE:DEC A:JR NZ,SEISU4
367D EB                  518              EX DE,HL
367E C1 C9               519             POP BC:RET
3680                     520 ;********
3680                     521 MEMOVR  ;メモリーオーバー表示
3680 2A 2F 30 2E 00 CD 1E 522            LD HL,(scrnsiz):LD L,0:CALL #LOC
3687 20
3688 CD 8E 36 C3 10 38   523             CALL MEMOVR1:JP   PDATA
368E 11 CB 31 CD E5 1F   524 MEMOVR1 LD DE,memovr:CALL #MSX
3694 CD C4 1F 18 06      525             CALL #BELL:JR ERROR1
3699                     526 ;
3699                     527 ERROR   ; エラー表示
3699                     528
3699 B7 28 F2            529             IF A=0 JR MEMOVR1
369C CD 33 20            530             CALL #ERROR
369F AF 32 71 30 32 80 30 531 ERROR1  XOR A:LD (inkymode),A:LD (chrbf),A
36A6 C3 21 20            532             JP   #FLGET
36A9                     533 ;-----------------------
36A9                     534 INKEYSUB
36A9                     535             ;マクロ中ならINKEY$へ
36A9                     536             ;  なければ 1文字入力
36A9 3A 72 30            537             LD A,(mcroflog)
36AC B7 CA 21 20         538             IF A=0 JP #FLGET
36B0                     539
36B0                     540 INKEY$  ;out Acc=ASCII
36B0                     541             ;       C=control key
36B0                     542             ;      NC=normal  key
36B0                     543             ;des nothing
36B0                     544
36B0 E5 D5 C5            545             PUSH HL:PUSH DE:PUSH BC
36B3 3A 71 30            546              LD A,(inkymode)
36B6 3D 20 05 CD DF 36 18 547             DEC A:IF Z THEN CALL COPYGET:JR INKEY$1
36BD 0B
36BE 3D 20 05 CD CD 36 18 548             DEC A:IF Z THEN CALL DELGET :JR INKEY$1
36C5 03
36C6 CD 44 37            549              CALL FLGET
36C9 C1 D1 E1 C9         550 INKEY$1 POP BC:POP DE:POP HL:RET
36CD                     551
36CD                     552 ;********
```

▶ところで，7月号のSTUDIO Xのタイトル横のイラストは，夏になったらシャープが水中でも使えるコンピュータを開発して，全国の川に放流するということを伝えようとしているんですか。もしそれが本当なら，この場合のエサはなにを付ければいいんでしょ。

林　保礼（16）大阪府

```
36CD                    553 DELGET  ; デリートバッファからの入力
36CD                    554
36CD 2A 78 30 7C B5 28 6C  555          LD HL,(delpnt):IF HL=0 JR FLGET-4
36D4 2B 22 78 30        556          DEC HL:LD (delpnt),HL
36D8 ED 5B 29 30 19     557          LD DE,(delbf):ADD HL,DE
36DD 18 08              558          JR COPYGET1
36DF                    559 ;********
36DF                    560 COPYGET ; コピーバッファからの入力
36DF                    561
36DF 2A 76 30 23        562          LD HL,(copypnt):INC HL
36E3 22 76 30 2B        563          LD (copypnt),HL:DEC HL
36E7 CD 94 1F B7 28 53  564 COPYGET1 CALL #PEEK:IF A=0 JR FLGET-4
36ED 4F FE 20 D2 65 37  565          LD C,A:IF A>=20H JP FLGETNO
36F3 C3 67 37           566          JP FLGETCO
36F6                    567 ;********
36F6 CD D8 1F FE 1B 28 1B  568 KMACRO  CALL #GETKY:IF A=1BH JR KMACRO1
36FD 2A 7C 30 7E 4F     569          LD HL,(macrpnt):LD A,(HL):LD C,A
3702 23 22 7C 30        570          INC HL:LD (macrpnt),HL
3706 B7 28 0F           571          IF A=0 JR KMACRO1
3709 FE 45 20 11        572          IF A<>"E" JR KMACRO2
370D 7E                 573                 LD A,(HL)
370E FE 4E 20 0C        574          IF A<>"N" JR KMACRO2
3712 23 7E              575          INC HL:LD A,(HL)
3714 FE 44 20 06        576          IF A<>"D" JR KMACRO2
3718 AF 32 72 30        577 KMACRO1 XOR A:LD (mcroflog),A
371C 18 26              578          JR FLGET
371E 79 FE 20 38 D3     579 KMACRO2 LD A,C:IF A<20H JR KMACRO
3723 FE 40 20 2C        580          IF A<>"@" JR FLGET1
3727 2A 7C 30 7E 4F     581          LD HL,(macrpnt):LD A,(HL):LD C,A
372C 23 22 7C 30        582          INC HL:LD (macrpnt),HL
3730 FE 20 38 C2        583          IF A<20H JR KMACRO
3734 FE 40 28 1B        584          IF A="@" JR FLGET1
3738 3E 1B 32 80 30     585          LD A,1BH:LD (chrbf),A
373D 79 18 13           586          LD A,C:JR FLGET1
3740                    587 ;********
3740 AF 32 71 30        588          XOR A:LD (inkymode),A
3744                    589 FLGET   ; キーボードからの入力
3744 3A 72 30           590          LD A,(mcroflog)
3747 B7 20 AC           591          IF A<>0 JR KMACRO
374A CD B5 37           592          CALL PCURDAT
374D CD A0 37 CD 21 20  593          CALL INKEYCUR:CALL #FLGET
3753 4F FE 20 38 1A     594 FLGET1  LD C,A:IF A<20H JR FLGET_c
3758 3A 74 30 B7 20 34  595          LD A,(ctrlkeep):IF A<>0  JR FLGET_e
375E 3A 80 30 FE 1B 28 2D  596          LD A,(chrbf)  :IF A=1BH JR FLGET_e
3765 B7 3E              597 FLGETNO OR A:DB 3EH    ;LD A,n ;normal key
3767 37                 598 FLGETCO SCF                    ;control key
3768 79 32 80 30        599          LD A,C:LD (chrbf),A
376C F5                 600          PUSH AF
376D CD 79 38           601           CALL PNORMK
3770 F1 C9              602          POP AF:RET
3772                    603          ;
3772 21 81 30 06 00     604 FLGET_c  LD HL,keyconf0:LD B,0
3777 09 7E 4F           605           ADD HL,BC:LD A,(HL):LD C,A
377A FE 1B 20 E9        606           IF A<>1BH JR FLGETCO
377E 21 80 30 BE 28 E3  607           LD HL,chrbf:IF A=(HL) JR FLGETCO
3784 32 80 30 3A 74 30  608           LD (chrbf),A:LD A,(ctrlkeep)
378A B7 20 DA           609           IF A<>0 JR FLGETCO
378D CD 8D 38 18 B2     610           CALL PCTRLK:JR FLGET
3792                    611          ;
3792 79 FE 40 38 DB     612 FLGET_e  LD A,C:IF A<40H  JR FLGET_c
3797 FE 60 30 CA        613                   IF A>=60H JR FLGETNO
379B D6 40 4F 18 D2     614           SUB 40H:LD C,A:JR FLGET_c
37A0                    615
37A0                    616 ;********
37A0                    617 INKEYCUR ; 1文字入力時のカーソルセット
37A0 3A 2E 30 4F        618          LD A,(l_offs):LD C,A
37A4 3A 7E 30 91 4F     619          LD A,(pointer):SUB C:LD C,A
37A9 3A 32 30 81 6F     620          LD A,(xstart):ADD A,C:LD L,A
37AE 3A 2D 30 67 C3 1E 20  621          LD A,(l_cury):LD H,A:JP #LOC
37B5                    622 ;************************
37B5                    623 PCURDAT ; カーソルデータ
37B5                    624          ; des HL,DE,BC
37B5                    625
37B5 2A 32 30 25        626          LD HL,(xstart):DEC H
37B9 3A 34 30 D6 0E     627          LD A,(xsize):SUB 14
37BE 85 6F CD 1E 20     628          ADD A,L:LD L,A:CALL #LOC
37C3 2A 7E 30 26 00     629          LD HL,(pointer):LD H,0
37C8 CD 39 36 11 0C 31 C3  630          CALL SEISU:LD DE,namare+2:JP #MSX
37CF E5 1F
37D1                    631 ;********
37D1                    632 PLINDAT ; 行Noデータ
37D1                    633          ; des HL,DE,BC
37D1                    634
37D1 2A 32 30 25        635          LD HL,(xstart):DEC H
37D5 3A 34 30 D6 07     636          LD A,(xsize):SUB 7
37DA 85 6F CD 1E 20     637          ADD A,L:LD L,A:CALL #LOC
37DF 2A 5D 30 CD 39 36  638          LD HL,(lineno):CALL SEISU
37E5 11 0A 31 C3 E5 1F  639          LD DE,namare:JP #MSX
37EB                    640 ;********
37EB                    641 PWINDNO ; ウィンドゥNo表示
37EB                    642          ; des HL,DE
37EB                    643
37EB 2A 32 30 ED 5B 34 30  644          LD HL,(xstart):LD DE,(xsize)
37F2 19 3E F6           645          ADD HL,DE:LD A,0-10
37F5 85 6F CD 1E 20     646          ADD A,L:LD L,A:CALL #LOC
37FA 11 56 31 CD E5 1F  647          LD DE,windm:CALL #MSX
3800 3A 31 30 C6 30 CD F4  648          LD A,(windno):ADD A,"0":CALL #PRINT
3807 1F
3808 CD F1 1F 3E 3E C3 F4  649          CALL #PRNTS:LD A,">":JP  #PRINT
380F 1F
3810                    650
3810                    651 ;********
3810                    652 PDATA   ; 各データ表示
3810                    653          ; des HL,DE,BC
3810                    654
3810 CD 40 38 CD 62 38 CD  655          CALL PTXTDAT:CALL PTYPDAT:CALL PNORMK
3817 79 38
3819 2A 2F 30           656          LD HL,(scrnsiz)
381C 7D D6 0A 6F CD 1E 20  657          LD A,L:SUB 10:LD L,A:CALL #LOC
3823 11 17 31 CD E5 1F  658          LD DE,data2m:CALL #MSX
3829                    659 ;********
3829                    660 PLEFDAT ; 残りバイト数データ
3829                    661          ; des HL,DE,BC
3829                    662
3829 2A 2F 30           663          LD HL,(scrnsiz)
382C 7D D6 08 6F CD 1E 20  664          LD A,L:SUB 8:LD L,A:CALL #LOC
3833 CD 06 36 D8        665          CALL LEFBYT:RET C
3837 CD 3F 36           666          CALL SEISU0
383A 11 0A 31 C3 E5 1F  667          LD DE,namare:JP #MSX
3840                    668 ;********
3840                    669 PTXTDAT ; 現在の ウィンドウ/テキスト Noデータ
3840                    670          ; des HL,DE,BC
3840                    671
3840 2A 2F 30           672          LD HL,(scrnsiz)
3843 2E 00 CD 1E 20     673          LD L,0:CALL #LOC
3848 3E 23 CD F4 1F     674          LD A,"#":CALL #PRINT
384D 3A 31 30 C6 30 CD F4  675          LD A,(windno):ADD A,"0":CALL #PRINT
3854 1F
3855 3E A5 CD F4 1F     676          LD A,"･":CALL #PRINT
385A 3A 5C 30 C6 41 C3 F4  677          LD A,(nowtxt):ADD A,"A":JP  #PRINT
3861 1F
3862                    678 ;********
3862                    679 PTYPDAT ; タイプモードデータ
3862                    680          ; des HL,DE
3862                    681
3862 2A 2F 30           682          LD HL,(scrnsiz)
3865 2E 04 CD 1E 20     683          LD L,4:CALL #LOC
386A 11 21 31 3A 73 30  684          LD DE,insertm:LD A,(insmode)
3870 B7 20 03 11 35 31  685          IF A=0 THEN LD DE,overtpm
3876 C3 E5 1F           686          JP #MSX
3879                    687 ;********
3879                    688 PNORMK  ; ノーマルキー モード 表示
3879                    689          ; des HL,DE
3879 3A 74 30 B7 20 0E  690          LD A,(ctrlkeep):IF A<>0 JR PCTRLK
387F 2A 2F 30           691          LD HL,(scrnsiz)
3882 2E 17 CD 1E 20     692          LD L,23:CALL #LOC
3887 11 49 31 C3 E5 1F  693          LD DE,narmalm:JP #MSX
388D                    694          ;
388D                    695 PCTRLK  ; コントロールキー モード 表示
388D                    696          ; des HL,DE
388D 2A 2F 30           697          LD HL,(scrnsiz)
3890 2E 17 CD 1E 20     698          LD L,23:CALL #LOC
3895 3A 74 30 26 2A     699          LD A,(ctrlkeep):LD H,"*"
389A B7 20 02 26 20     700          IF A=0 THEN LD H," "
389F 7C CD F4 1F        701          LD A,H:CALL #PRINT
38A3 11 50 31 C3 E5 1F  702          LD DE,ctrlm:JP  #MSX
38A9                    703
38A9                    704 ;----------------------
38A9                    705 PAGE2   ;next page LABEL

0000                      1 ;----------------------------
0000                      2 ;    WINER          page 2
0000                      3 ;       < screen editer >
0000                      4 ;    Copyright in 1988 by 環
0000                      5 ;----------------------------
0000                      6
0000                      7          OFFSET  makadr-program
38A9                      8          ORG     PAGE2
38A9                      9
38A9                     10 ;************************
38A9                     11 GETL    ; 1行入力
38A9                     12          ; in/out noting /des all
38A9                     13          ; set line buff
38A9                     14          ;  at break (linebf)=1BH
38A9                     15
38A9 AF                  16          XOR A
38AA 32 7F 30 32 80 30   17          LD (inpchk),A:LD (chrbf),A
38B0 CD A9 34            18          CALL TABPRT
38B3 ED 5B 10 30 CD 03 4C  19 GETL1  LD DE,(linebf):CALL LNPRINT
38BA CD B0 36 38 1B      20 GETL2  CALL INKEY$:JR C,GETL0D
38BF FE 20 38 F7         21          IF A<20H JR GETL2
38C3 08                  22          EX AF,AF'
38C4 3A 73 30            23           LD A,(insmode)
38C7 B7 C4 4C 3C         24           IF A<>0 CALL INSCHR
38CB CD C4 39            25           CALL PINADR
38CE 08                  26          EX AF,AF'
38CF 77 CD D2 39         27          LD (HL),A:CALL CURR
38D3 3E 01 32 7F 30      28          LD A,1:LD (inpchk),A
38D8 18 D9               29          JR GETL1
38DA                     30          ;
38DA FE 0D 20 0F         31 GETL0D  IF A<>0DH JR GETL1B
38DE AF 32 7E 30 32 2E 30  32        XOR A:LD (pointer),A:LD (l_offs),A
38E5 3C 32 7F 30 C9      33          INC A:LD (inpchk),A:RET
38EA C3 03 4C            34          JP LNPRINT
38ED                     35          ;
38ED FE 1B 20 06         36 GETL1B  IF A<>1BH JR GETLCT ;break
38F1 12                  37          LD (DE),A
38F2 AF 32 7F 30         38          XOR A:LD (inpchk),A
38F6 C9                  39          RET
38F7                     40          ;
38F7 CD 00 39 C0         41 GETLCT  CALL CTRLSEL:RET NZ
38FB CD 09 39 18 B3      42          CALL CTRLJP:JR GETL1
3900                     43 ;********
3900                     44 CTRLSEL ; 1行入力時に有効な
3900                     45          ; コントロールキーを選択
3900                     46          ; in  Acc=ctrl key
3900                     47          ; out flog Z=有効
3900                     48          ; des HL,BC
3900                     49
3900 21 17 39            50          LD HL,ctrlsel
3903 01 24 00            51          LD BC,ctrljp-ctrlsel
3906 ED B1 C9            52          CPIR:RET
3909                     53 ;
3909                     54 CTRLJP  ;コントロールキージャンプ 実行
3909                     55          ; in  Acc=ctrl key
3909                     56          ; des all
3909                     57
3909 87 21 3B 39         58          ADD A,A:LD HL,ctrljp
390D 85 6F 30 01 24      59          ADD A,L:LD L,A:IF C THEN INC H
3912 7E 23 66            60          LD A,(HL):INC HL:LD H,(HL)
3915 6F E9               61          LD L,A:JP (HL)
3917                     62 ;
3917 01 02               63 ctrlsel DB    01H,02H
3919 04 06 07            64          DB 04H,    06H,07H
391C 08 09 0B            65          DB 08H,09H    ,0BH
391F 0F                  66          DB            0FH
3920 10 11 13            67          DB 10H,11H,    13H
3923 16                  68          DB        16H
3924 1C 1D               69          DB 1CH,1DH
3926 20                  70          DB 20H
3927 2A 2B               71          DB         2AH,2BH
3929 2C 2E 2F            72          DB 2CH,    2EH,2FH
392C 30 31 32 33         73          DB 30H,31H,32H,33H
3930 34 35 36 37         74          DB 34H,35H,36H,37H
3934 38 39 3A 3B         75          DB 38H,39H,3AH,3BH
3938 3C 3E 3F            76          DB 3CH,    3EH,3FH
393B                     77          ;
393B C3 39 4C 3A A7 3A BF  78 ctrljp  DW NONOP:WORDL:BAKSP:PAGD  ;00H,01H,02H,03H ; @ A B C
3942 3D
3943 D2 39 B7 3C 22 3A B0  79        DW CURR :CURU :WORDR:DEL    ;04H,05H,06H,07H ; D E F G
394A 3A
394B BB 39 73 3A B7 3D C7  80        DW NOWRT:TABU :DRETR:DBFCL  ;08H,09H,0AH,0BH ; H I J K
3952 3B
3953 AF 3E 71 3C 98 3E 41  81        DW DELCR:CR   :INSCR:INSMD  ;0CH,0DH,0EH,0FH ; L M N O
395A 3C
395B                     82
395B C1 3B 8B 3A EB 3D 08  83        DW DELBP:TABST:PAGU :CURL   ;10H,11H,12H,13H ; P Q R S
3962 3A
3963 8F 3D 6E 3D E0 3A 2F  84        DW DSPL :DSPR :CAFTR:DSPU   ;14H,15H,16H,17H ; T U V W
396A 3D
396B 78 3C 49 3E FF 3C C3  85        DW CURD :CLINE:DSPD :NONOP  ;18H,19H,1AH,1BH ; X Y Z ESC
3972 39
3973 D2 39 08 3A B7 3C 78  86        DW CURR :CURL :CURU :CURD   ;1CH,1DH,1EH,1FH ;
397A 3C
397B                     87
397B 66 3C FB 3F 34 40 5A  88        DW CTKEP:JPWIN:WINMX:WINMV  ;20H,21H,22H,23H ;   ! " #
3982 40
3983 9D 40 E7 3E 62 3F 09  89        DW WINSZ:MARK :CUTXT:GETXT  ;24H,25H,26H,27H ; $ % & '
398A 3F
398B 9E 3F C3 39 E6 3B E3  90        DW PUTXT:NONOP:MACR6:MACR5  ;28H,29H,2AH,2BH ; ( ) * +
3992 3B
3993 DA 3B C3 39 DD 3B E0  91        DW MACR2:NONOP:MACR3:MACR4  ;2CH,2DH,2EH,2FH ; , - . /
399A 3B
399B                     92
399B 34 3B A8 3B AA 3B AD  93        DW CPYDS:CPYP1:CPYP2:CPYP3  ;30H,31H,32H,33H ; 0 1 2 3
39A2 3B
39A3 73 3B 75 3B 78 3B 81  94        DW CPYC1:CPYC2:CPYC3:CPYG1  ;34H,35H,36H,37H ; 4 5 6 7
39AA 3B
39AB 83 3B 86 3B D7 3B D4  95        DW CPYG2:CPYG3:MACR1:MACR0  ;38H,39H,3AH,3BH ; 8 9 : ;
39B2 3B
39B3 E9 3B C3 39 EC 3B EF  96        DW MACR7:NONOP:MACR8:MACR9  ;3CH,3DH,3EH,3FH ; < = > ?
39BA 3B
39BB                     97
39BB                     98 ;************************
39BB                     99 NOWRT   ;入力キャンセル
39BB AF 32 7F 30 32 80 30  100        XOR A:LD (inpchk),A:LD (chrbf),A
39C2 E1                 101          POP HL
39C3                    102 ;--------
39C3                    103 NONOP   ;何もしない
39C3 C9                 104          RET
39C4                    105 ;--------
39C4                    106 PINADR ;ポインターアドレスを返す
```

▶7月号のSTUDID Xに「ソーサリアンである～(以下繰り返し)」が載ったということは，当然,「イースIIである～」も載るということで，であれば「ハイドライド3である～」というのもあり得るのである。というわけで(どこがぢゃ)，私も載りたい。

宮崎　和也（17）佐賀県

```
39C4                    107              ; out HL=pointer adress  A=pointer
39C4 3A 7E 30 2A 10 30  108              LD A,(pointer):LD HL,(linebf)
39CA F5                 109              PUSH AF
39CB 85 6F              110               ADD A,L:LD L,A
39CD 30 01 24           111               IF C THEN INC H
39D0 F1                 112              POP AF
39D1 C9                 113              RET
39D2                    114 ;------------------------
39D2                    115 CURR   ;１文字右
39D2 3A 7E 30 3C        116              LD A,(pointer):INC A
39D6 FE FE 38 02 3E FE  117              IF A>=254 THEN LD A,254
39DC 32 7E 30 D0        118              LD (pointer),A:RET NC
39E0                    119 ;--------
39E0                    120 OFFSTR ;右移動時のオフセット
39E0 CD FB 39 16 01     121              CALL OFFSTR3:LD D,1
39E5 7C 90              122 OFFSTR1 LD A,H:SUB B
39E7 B9 38 08           123              IF A<C JR OFFSTR2
39EA 78 C6 0A 47        124              LD A,B:ADD A,10:LD B,A
39EE 16 00 18 F3        125              LD D,0:JR OFFSTR1
39F2 78 32 2E 30        126 OFFSTR2 LD A,B:LD (l_offs),A
39F6 15 C8              127              IF DEC(D)=0 RET
39F8 C3 A9 34           128              JP TABPRT
39FB                    129              ;
39FB 3A 34 30 4F        130 OFFSTR3 LD A,(xsize):LD C,A
39FF 3A 2E 30 47        131              LD A,(l_offs):LD B,A
3A03 3A 7E 30 67 C9     132              LD A,(pointer):LD H,A:RET
3A08                    133 ;--------
3A08                    134 CURL   ;１文字左
3A08 3A 7E 30 B7 C8     135              LD A,(pointer):IF A=0 RET
3A0D 3D 32 7E 30        136              DEC A:LD (pointer),A
3A11                    137 ;--------
3A11                    138 OFFSTL ;左移動時のオフセット
3A11 CD FB 39 16 01     139              CALL OFFSTR3:LD D,1
3A16 7C B8 30 D8        140 OFFSTL1 LD A,H:IF A>=B JR OFFSTR2
3A1A 78 D6 0A 47        141              LD A,B:SUB 10:LD B,A
3A1E 16 00 18 F4        142              LD D,0:JR OFFSTL1
3A22                    143 ;--------
3A22                    144 WORDR  ;１単語右
3A22 CD C4 39 4F        145              CALL PINADR:LD C,A
3A26 7E FE 0D C8        146 WORDR1  LD A,(HL):IF A=0DH RET
3A2A CD F9 35 28 04     147              CALL ?SKIP:JR Z,WORDR2
3A2F 23 0C 18 F3        148              INC HL:INC C:JR WORDR1
3A33 79 32 7E 30        149 WORDR2  LD A,C:LD (pointer),A
3A37 7E FE 0D 28 0D     150              LD A,(HL):IF A=0DH JR WORDR3+4
3A3C CD F9 35 20 04     151              CALL ?SKIP:JR NZ,WORDR3
3A41 23 0C 18 F2        152              INC HL:INC C:JR WORDR2+4
3A45 79 32 7E 30        153 WORDR3  LD A,C:LD (pointer),A
3A49 C3 E0 39           154              JP OFFSTR
3A4C                    155 ;--------
3A4C                    156 WORDL  ;１単語左
3A4C CD C4 39 4F B7 C8  157              CALL PINADR:LD C,A:IF A=0 RET
3A52 2B 0D 28 16        158              DEC HL:IF DEC(C)=0 JR WORDL3
3A56 7E 2B              159 WORDL1  LD A,(HL):DEC HL
3A58 CD F9 35 20 05     160              CALL ?SKIP:JR NZ,WORDL2
3A5D 0D 20 F6           161              IF DEC(C)<>0 JR WORDL1
3A60 18 0A              162              JR WORDL3
3A62 7E 2B              163 WORDL2  LD A,(HL):DEC HL
3A64 CD F9 35 28 03     164              CALL ?SKIP:JR Z,WORDL3
3A69 0D 20 F6           165              IF DEC(C)<>0 JR WORDL2
3A6C 79 32 7E 30        166 WORDL3  LD A,C:LD (pointer),A
3A70 C3 11 3A           167              JP OFFSTL
3A73                    168 ;------------------------
3A73                    169 TABU   ;タブ出力
3A73 CD 9A 3A           170              CALL TABUST1
3A76 23 0C 7E           171 TABU1   INC HL:INC C:LD A,(HL)
3A79 FE 0D 20 03 0D 18 04 172            IF A=0DH THEN DEC C:JR TABU2
3A80 FE 2B 20 F2        173              IF A<>"+" JR TABU1
3A84 79 32 7E 30        174 TABU2   LD A,C:LD (pointer),A
3A88 C3 E0 39           175              JP OFFSTR
3A8B                    176 ;--------
3A8B                    177 TABST  ;タブセット
3A8B CD 9A 3A           178              CALL TABUST1
3A8E 7E 0E 2B           179              LD A,(HL):LD C,"+"
3A91 B9 20 02 0E 2D     180              IF A=C THEN LD C,"-"
3A96 71 C3 A9 34        181              LD (HL),C:JP TABPRT
3A9A 2A 0E 30 3A 7E 30  182 TABUST1 LD HL,(tabadr):LD A,(pointer)
3AA0 4F 85 6F           183              LD C,A:ADD A,L:LD L,A
3AA3 30 01 24           184              IF C THEN INC H
3AA6 C9                 185              RET
3AA7                    186 ;------------------------
3AA7                    187 BAKSP  ;カーソルの左１文字を削除
3AA7 3A 7E 30 B7 C8     188              LD A,(pointer):IF A=0 RET
3AAC 3D 32 7E 30        189              DEC A:LD (pointer),A
3AB0                    190 ;--------
3AB0                    191 DEL    ;カーソル上の１文字を削除
3AB0 CD C4 39 FE FE 28 22 192            CALL PINADR:IF A=254 JR DEL1
3AB7 4F 3E FE 91        193              LD C,A:LD A,254:SUB C
3ABB 4F 06 00           194              LD C,A:LD B,0
3ABE EB                 195              EX DE,HL
3ABF 2A 78 30 23        196              LD HL,(delpnt):INC HL
3AC3 1A CD 1D 3B        197              LD A,(DE):CALL DELPUSH
3AC7 23 22 78 30        198              INC HL:LD (delpnt),HL
3ACB AF CD 1D 3B        199              XOR A:CALL DELPUSH
3ACF 62 6B 23           200              LD HL,DE:INC HL
3AD2 0C 0D 28 02 ED B0  201              IF C<>0 THEN LDIR
3AD8 2B                 202              DEC HL
3AD9 36 20              203 DEL1    LD (HL)," "
3ADB 3C 32 7F 30        204              INC A:LD (inpchk),A
3ADF C9                 205              RET
3AE0                    206 ;--------
3AE0                    207 CAFTR  ;カーソル以後を削除
3AE0 CD 08 3B C8        208              CALL CAFTR2:RET Z
3AE4 3A 7E 30 4F 78     209              LD A,(pointer):LD C,A:LD A,B
3AE9 91 D8              210              SUB C:RET C
3AEB 3C 47              211              INC A:LD B,A
3AED 2A 78 30 23        212              LD HL,(delpnt):INC HL
3AF1 1A CD 1D 3B        213 CAFTR1  LD A,(DE):CALL DELPUSH
3AF5 3E 20 12           214              LD A," ":LD (DE),A
3AF8 1B 23 10 F5        215              DEC DE:INC HL:DJNZ CAFTR1
3AFC 22 78 30           216              LD (delpnt),HL
3AFF AF CD 1D 3B        217              XOR A:CALL DELPUSH
3B03 3C 32 7F 30        218              INC A:LD (inpchk),A
3B07 C9                 219              RET
3B08                    220              ;
3B08 06 FE 11 FE 00     221 CAFTR2  LD B,254:LD DE,254
3B0D 2A 10 30 19 EB     222              LD HL,(linebf):ADD HL,DE:EX DE,HL
3B12 1A FE 20 C0        223 CAFTR3  LD A,(DE):IF A<>" " RET
3B16 1B 10 F9           224              DEC DE:DJNZ CAFTR3
3B19 1A FE 20 C9        225              LD A,(DE):CP " ":RET
3B1D                    226
3B1D                    227 DELPUSH ;デリートバッファに
3B1D                    228              ; Accを積む
3B1D D5                 229              PUSH DE
3B1E 08 3A 2B 30        230               EX AF,AF':LD A,(delmsk)
3B22 A4 67              231               AND H:LD H,A
3B24 B5 20 01 2C        232               OR L:IF Z THEN INC L
3B28 EB CD CD 3B        233               EX DE,HL:CALL DBFCL1
3B2C 19                 234               ADD HL,DE
3B2D 08 CD 9A 1F        235               EX AF,AF':CALL #POKE
3B31 EB                 236               EX DE,HL
3B32 D1 C9              237              POP DE:RET
3B34                    238 ;------------------------
3B34                    239 ;   コピーバッファ表示
3B34 2A 2F 30 7D 2E 00  240 CPYDS   LD HL,(scrnsiz):LD A,L:LD L,0
3B3A D6 0C 47 0E 03 ED 5B 241            SUB 12:LD B,A:LD C,3:LD DE,(copybf)
3B41 1E 30
3B43 C5 D5 E5           242 CPYDS1  PUSH BC:PUSH DE:PUSH HL
3B46 CD 1E 20 EB        243              CALL #LOC:EX DE,HL
3B4A CD 94 1F 23        244 CPYDS2   CALL #PEEK:INC HL
3B4E FE 20 30 03 3E 20 2B 245            IF A<20H THEN LD A," ":DEC HL
3B55 CD F4 1F 10 F0     246              CALL #PRINT:DJNZ CPYDS2
3B5A E1 D1 C1           247              POP HL:POP DE:POP BC
3B5D 14 CD A9 36        248              INC D:CALL INKEYSUB
3B61 FE 20 20 03 0D 20 DB 249            IF A=" " THEN DEC C:JR NZ,CPYDS1
3B68 CD 1E 20 CD F1 1F 10 250 CPYDS3 CALL #LOC:CALL #PRNTS:DJNZ CPYDS3+3
3B6F FB
3B70 C3 10 38           251              JP PDATA
3B73                    252
3B73                    253 ;--------
3B73                    254 ;   コピーバッファクリア
3B73                    255
3B73 AF 01              256 CPYC1   XOR A   :DB 01H
3B75 3E 01 01           257 CPYC2   LD A,01H:DB 01H
3B78 3E 02 CD BB 3B     258 CPYC3   LD A,02H:CALL CPYP4
3B7D AF C3 9A 1F        259              XOR A:JP #POKE
3B81                    260 ;--------
3B81                    261 ;   コピーバッファ  入力
3B81 AF 01              262 CPYG1   XOR A   :DB 01H
3B83 3E 01 01           263 CPYG2   LD A,01H:DB 01H
3B86 3E 02 CD BB 3B     264 CPYG3   LD A,02H:CALL CPYP4
3B8B 06 FE              265              LD B,254
3B8D CD 94 1F           266 CPYG4   CALL #PEEK
3B90 B7 28 03 23 10 F7  267              IF A<>0 THEN INC HL:DJNZ CPYG4
3B96 C0                 268              RET NZ
3B97 EB CD C4 39 EB     269              EX DE,HL:CALL PINADR:EX DE,HL
3B9C 1A CD 9A 1F        270              LD A,(DE):CALL #POKE
3BA0 23 AF CD 9A 1F     271              INC HL:XOR A:CALL #POKE
3BA5 C3 D2 39           272              JP CURR
3BA8                    273 ;--------
3BA8                    274 ; コピーバッファ  出力
3BA8 AF 01              275 CPYP1   XOR A   :DB 01H
3BAA 3E 01 01           276 CPYP2   LD A,01H:DB 01H
3BAD 3E 02 CD BB 3B     277 CPYP3   LD A,02H:CALL CPYP4
3BB2 3E 01 32 71 30     278              LD A,1 :LD (inkymode),A
3BB7 22 76 30 C9        279              LD (copypnt),HL:RET
3BBB                    280              ;
3BBB 2A 1E 30 84        281 CPYP4   LD HL,(copybf):ADD A,H
3BBF 67 C9              282              LD H,A:RET
3BC1                    283 ;------------------------
3BC1                    284 DELBP  ;デリートバッファ出力
3BC1 3E 02 32 71 30     285              LD A,2:LD (inkymode),A
3BC6 C9                 286              RET
3BC7                    287 ;--------
3BC7                    288 DBFCL  ;デリートバッファクリア
3BC7 21 00 00 22 78 30  289              LD HL,0:LD (delpnt),HL
3BCD 2A 29 30 AF C3 9A 1F 290 DBFCL1 LD HL,(delbf):XOR A:JP #POKE
3BD4                    291 ;------------------------
3BD4                    292 ;マクロ実行
3BD4 3E 30 01           293 MACR0   LD A,"0" DB 01H
3BD7 3E 31 01           294 MACR1   LD A,"1" DB 01H
3BDA 3E 32 01           295 MACR2   LD A,"2" DB 01H
3BDD 3E 33 01           296 MACR3   LD A,"3" DB 01H
3BE0 3E 34 01           297 MACR4   LD A,"4" DB 01H
3BE3 3E 35 01           298 MACR5   LD A,"5" DB 01H
3BE6 3E 36 01           299 MACR6   LD A,"6" DB 01H
3BE9 3E 37 01           300 MACR7   LD A,"7" DB 01H
3BEC 3E 38 01           301 MACR8   LD A,"8" DB 01H
3BEF 3E 39              302 MACR9   LD A,"9"
3BF1                    303              ;
3BF1 32 40 3C           304              LD (MACRNO),A
3BF4 3E 03 CD CC 34 D8  305              LD A,3:CALL TEXTDAT:RET C
3BFA E5 E5              306              PUSH HL:PUSH HL
3BFC AF ED B1           307                XOR A:CPIR
3BFF C1                 308               POP BC
3C00 B7 ED 42 44 4D     309               SUB HL,BC:LD BC,HL
3C05 E1 D8 C8           310              POP HL:RET C:RET Z
3C08 7E B7 C8           311              LD A,(HL):IF A=0 RET
3C0B 3E 4D ED B1 C0     312 MACRO   LD A,"M":CPIR:RET NZ
3C10 7E FE 43 20 F6     313              LD A,(HL):IF A<>"C" JR MACRO
3C15 23 0B 78 B1 C8     314              INC HL:IF DEC(BC)=0 RET
3C1A 7E FE 52 20 EC     315              LD A,(HL):IF A<>"R" JR MACRO
3C1F 23 0B 78 B1 C8     316              INC HL:IF DEC(BC)=0 RET
3C24 3A 40 3C BE 20 E1  317              LD A,(MACRNO):IF A<>(HL) JR MACRO
3C2A 3E 0D              318              LD A,0DH
3C2C BE 23 20 FC        319 MACRO1  CP (HL):INC HL:JR NZ,MACRO1
3C30 22 7C 30           320              LD (macrpnt),HL
3C33 3E 01 32 72 30     321              LD A,1:LD (mcroflog),A
3C38 AF 3A 74 30        322              XOR A:LD A,(ctrlkeep)
3C3C 32 80 30 C9        323              LD (chrbf),A:RET
3C40 00                 324 MACRNO  DB 0
3C41                    325 ;------------------------
3C41                    326 INSMD  ;インサート／オーバーライト  モード切換え
3C41 3A 73 30 EE 01     327              LD A,(insmode):XOR 1
3C46 32 73 30 C3 62 38  328              LD (insmode),A:JP PTYPDAT
3C4C                    329 ;
3C4C                    330 INSCHR ;キャラクターインサート
3C4C 3A 7E 30 4F        331              LD A,(pointer):LD C,A
3C50 3E FE B9 C8        332              LD A,254:IF A=C RET
3C54 91 4F 06 00        333              SUB C:LD C,A:LD B,0
3C58 21 FE 00 19        334              LD HL,254:ADD HL,DE
3C5C 7E FE 20 C0        335              LD A,(HL):IF A<>" " RET
3C60 54 5D 2B ED B8 C9  336              LD DE,HL:DEC HL:LDDR:RET
3C66                    337 ;--------
3C66                    338 CTKEP  ;コントロールキー切換え
3C66 3A 74 30 EE 01     339              LD A,(ctrlkeep):XOR 1
3C6B 32 74 30 C3 79 38  340              LD (ctrlkeep),A:JP PNORMK
3C71                    341 ;********************************
3C71                    342 CR     ;キャリッジリターン
3C71 AF 32 7E 30 32 2E 30 343            XOR A:LD (pointer),A:LD (l_offs),A
3C78                    344 ;--------
3C78                    345 CURD   ;１文字下
3C78 CD 48 35 DA 80 36  346              CALL WRTLIN:JP C,MEMOVR
3C7E ED 5B 5F 30 1A B7 C8 347            LD DE,(lineadr):LD A,(DE):IF A=0 RET
3C85 CD 76 34 ED 53 5F 30 348            CALL NXTLIN:LD (lineadr),DE
3C8C 2A 5D 30 23 22 5D 30 349            LD HL,(lineno):INC HL:LD (lineno),HL
3C93 ED 5B 61 30 ED 4B 35 350            LD DE,(dtopno):LD BC,(ysize):LD B,0
3C9A 30 06 00
3C9D B7 ED 52 B7 ED 42 D8 351            SUB HL,DE:SUB HL,BC:RET C
3CA4 13 ED 53 61 30     352              INC DE:LD (dtopno),DE
3CA9 ED 5B 63 30 CD 76 34 353            LD DE,(dtopadr):CALL NXTLIN
3CB0 ED 53 63 30 C3 AA 33 354            LD (dtopadr),DE:JP TXTPRT
3CB7                    355 ;--------
3CB7                    356 CURU   ;１文字上
3CB7 CD 48 35 DA 80 36  357              CALL WRTLIN:JP C,MEMOVR
3CBD 2A 5D 30 7C B5 C8  358              LD HL,(lineno):IF HL=0 RET
3CC3 2B 22 5D 30        359              DEC HL:LD (lineno),HL
3CC7 ED 5B 5F 30 CD ED 3C 360            LD DE,(lineadr):CALL CURU1:LD (lineadr),DE
3CCE ED 53 5F 30
3CD2 ED 5B 61 30 B7 ED 52 361            LD DE,(dtopno):SUB HL,DE:RET NC
3CD9 D0
3CDA 1B ED 53 61 30     362              DEC DE:LD (dtopno),DE
3CDF ED 5B 63 30 CD ED 3C 363            LD DE,(dtopadr):CALL CURU1
3CE6 ED 53 63 30 C3 AA 33 364            LD (dtopadr),DE:JP TXTPRT
3CED CD 80 34           365 CURU1   CALL DNXTLIN
3CF0 E5                 366              PUSH HL
3CF1 2A 67 30 B7 ED 52  367               LD HL,(ntxtst):SUB HL,DE
3CF7 38 04 ED 5B 67 30  368               IF NC THEN LD DE,(ntxtst)
3CFD E1 C9              369              POP HL:RET
3CFF                    370 ;------------------------
3CFF                    371 DSPD   ;１行下表示
3CFF CD 48 35 DA 80 36  372              CALL WRTLIN:JP C,MEMOVR
3D05 ED 5B 63 30 1A B7 C8 373            LD DE,(dtopadr):LD A,(DE):IF A=0 RET
3D0C CD 76 34 ED 53 63 30 374            CALL NXTLIN:LD (dtopadr),DE
3D13 ED 4B 61 30 03 ED 43 375            LD BC,(dtopno):INC BC:LD (dtopno),BC
3D1A 61 30
3D1C 2A 5D 30 B7 ED 42  376              LD HL,(lineno):SUB HL,BC
3D22 30 08 ED 43 5D 30 ED 377            IF C THEN LD (lineno),BC:LD (lineadr),DE
3D29 53 5F 30
3D2C C3 AA 33           378              JP TXTPRT
3D2F                    379 ;--------
3D2F                    380 DSPU   ;１行上表示
3D2F CD 48 35 DA 80 36  381              CALL WRTLIN:JP C,MEMOVR
```

THE SENTINEL

▶私はジャンボーグＡよりも，ジャンボーグナインのほうが好きだ。やはりハンドル操作が魅力である。ところでOh！Xの読者にはザブングルのファンはいないのだろうか。
山田　俊英（20）東京都

```
3D35 2A 61 30 7C B5 C8      382             LD HL,(dtopno):IF HL=0 RET
3D3B 2B 22 61 30            383             DEC HL:LD (dtopno),HL
3D3F ED 5B 63 30 CD ED 3C   384             LD DE,(dtopadr):CALL CURU1:LD (dtopadr),DE
3D46 ED 53 63 30
3D4A ED 4B 35 30 06 00 09   385             LD BC,(ysize):LD B,0:ADD HL,BC
3D51 ED 4B 5D 30 B7 ED 42   386             LD BC,(lineno):SUB HL,BC:JP NZ,TXTPRT
3D58 C2 AA 33
3D5B 0B ED 43 5D 30         387             DEC BC:LD (lineno),BC
3D60 ED 5B 5F 30 CD ED 3C   388             LD DE,(lineadr):CALL CURU1
3D67 ED 53 5F 30 C3 AA 33   389             LD (lineadr),DE:JP TXTPRT
3D6E                        390 ;--------
3D6E                        391 DSPR   ;スクリーン右
3D6E CD 48 35 DA 80 36      392             CALL WRTLIN:JP C,MEMOVR
3D74 3A 2E 30 C6 10         393             LD A,(l_offs):ADD A,16
3D79 38 03 32 2E 30         394             IF NC THEN LD (l_offs),A
3D7E 3A 2E 30 47 3A 7E 30   395             LD A,(l_offs):LD B,A:LD A,(pointer)
3D85 B8 30 04 78 32 7E 30   396             IF A<B THEN LD A,B:LD (pointer),A
3D8C C3 AA 33               397             JP TXTPRT
3D8F                        398 ;--------
3D8F                        399 DSPL   ;スクリーン左
3D8F CD 48 35 DA 80 36      400             CALL WRTLIN:JP C,MEMOVR
3D95 3A 2E 30 D6 10         401             LD A,(l_offs):SUB 16
3D9A 30 01 AF               402             IF C THEN XOR A
3D9D 32 2E 30 47            403             LD (l_offs),A:LD B,A
3DA1 3A 34 30 80 DA AA 33   404             LD A,(xsize):ADD A,B:JP C,TXTPRT
3DA8 47 3A 7E 30            405             LD B,A:LD A,(pointer)
3DAC B8 38 05 05 78 32 7E   406             IF A>=B THEN DEC B:LD A,B:LD (pointer),A
3DB3 30
3DB4 C3 AA 33               407             JP TXTPRT
3DB7                        408 ;--------
3DB7                        409 DRETR  ;スクリーンを表示しなおす
3DB7 CD 48 35 DA 80 36      410             CALL WRTLIN:JP C,MEMOVR
3DBD 18 F5                  411             JR DRETR-3
3DBF                        412 ;--------
3DBF                        413 PAGD   ;1ページ下表示
3DBF CD 48 35 DA 80 36      414             CALL WRTLIN:JP C,MEMOVR
3DC5 2A 61 30 ED 5B 63 30   415             LD HL,(dtopno):LD DE,(dtopadr)
3DCC 3A 35 30 47            416             LD A,(ysize):LD B,A
3DD0 1A B7 28 06            417 PAGD1  LD A,(DE):IF A=0 JR PAGD2
3DD4 CD 76 34 23 10 F6      418             CALL NXTLIN:INC HL:DJNZ PAGD1
3DDA 22 61 30 ED 53 63 30   419 PAGD2  LD (dtopno),HL:LD (dtopadr),DE
3DE1 22 5D 30 ED 53 5F 30   420             LD (lineno),HL:LD (lineadr),DE
3DE8 C3 AA 33               421             JP TXTPRT
3DEB                        422 ;--------
3DEB                        423 PAGU   ;1ページ上表示
3DEB CD 48 35 DA 80 36      424             CALL WRTLIN:JP C,MEMOVR
3DF1 2A 61 30 ED 5B 63 30   425             LD HL,(dtopno):LD DE,(dtopadr)
3DF8 7C B5 C8               426             IF HL=0 RET
3DFB ED 4B 35 30 06 00      427             LD BC,(ysize):LD B,0
3E01 B7 ED 42 30 09         428             SUB HL,BC:JR NC,PAGU1
3E06 21 00 00 ED 5B 67 30   429             LD HL,0:LD DE,(ntxtst):JR PAGU1+6
3E0D 18 06
3E0F CD ED 3C 0D 20 FA      430 PAGU1  CALL CURU1:DEC C:JR NZ,PAGU1
3E15 22 61 30 ED 53 63 30   431             LD (dtopno),HL:LD (dtopadr),DE
3E1C E5 D5                  432             PUSH HL:PUSH DE
3E1E ED 4B 35 30 06 00 09   433              LD BC,(ysize):LD B,0:ADD HL,BC
3E25 ED 5B 5D 30 B7 ED 52   434              LD DE,(lineno):SUB HL,DE
3E2C D1 E1                  435             POP DE:POP HL
3E2E 38 03 C2 AA 33         436             IF NC THEN JP NZ,TXTPRT
3E33 0D 28 09               437             IF DEC(C)=0 JR PAGU3
3E36 CD 76 34               438 PAGU2  CALL NXTLIN
3E39 28 04 23 0D 20 F7      439             IF NZ THEN INC HL:DEC C:JR NZ,PAGU2
3E3F 22 5D 30 ED 53 5F 30   440 PAGU3  LD (lineno),HL:LD (lineadr),DE
3E46 C3 AA 33               441             JP TXTPRT
3E49                        442 ;------------------------
3E49                        443 CLINE  ;カーソル上の1行を削除
3E49 CD 08 3B               444             CALL CAFTR2
3E4C F5                     445             PUSH AF
3E4D 2A 78 30 23            446             LD HL,(delpnt):INC HL
3E51 3E 0D CD 1D 3B 23      447             LD A,0DH:CALL DELPUSH:INC HL
3E57 3E 0E CD 1D 3B 23      448             LD A,0EH:CALL DELPUSH:INC HL
3E5D F1 28 09               449             POP AF:JR Z,CLINE2
3E60 04                     450             INC B
3E61 1A CD 1D 3B            451 CLINE1 LD A,(DE):CALL DELPUSH
3E65 1B 23 10 F8            452             DEC DE:INC HL:DJNZ CLINE1
3E69 AF CD 1D 3B 22 78 30   453 CLINE2 XOR A:CALL DELPUSH:LD (delpnt),HL
3E70 ED 5B 5F 30            454             LD DE,(lineadr)
3E74 D5                     455             PUSH DE
3E75 CD 76 34 62 6B         456              CALL NXTLIN:LD HL,DE
3E7A D1                     457             POP DE
3E7B 7E B7 28 10            458             LD A,(HL):IF A=0 JR CLINE3
3E7F E5                     459             PUSH HL
3E80 44 4D 2A 65 30         460              LD BC,HL:LD HL,(endadr)
3E85 B7 ED 42 44 4D         461              SUB HL,BC:LD BC,HL
3E8A E1                     462             POP HL
3E8B 03 ED B0 1B            463             INC BC:LDIR:DEC DE
3E8F AF 12 ED 53 65 30      464 CLINE3 XOR A:LD (DE),A:LD (endadr),DE
3E95 C3 AA 33               465             JP TXTPRT
3E98                        466 ;--------
3E98                        467 INSCR  ;カーソル位置にCRを書きこむ
3E98 CD 4C 3C CD C4 39      468             CALL INSCHR:CALL PINADR
3E9E 36 0D CD 4D 35 DA 80   469             LD (HL),0DH:CALL WRTLIN0:JP C,MEMOVR
3EA5 36
3EA6 CD 26 35 CD 4D 35      470             CALL LINTRS:CALL WRTLIN0
3EAC C3 AA 33               471             JP TXTPRT
3EAF                        472 ;--------
3EAF                        473 DELCR  ;カーソル行と次の行を連結
3EAF CD 4D 35 DA 80 36      474             CALL WRTLIN0:JP C,MEMOVR
3EB5 2A 5F 30 54 5D         475             LD HL,(lineadr):LD DE,HL
3EBA CD 76 34 C8            476             CALL NXTLIN:RET Z
3EBE 1A B7 C8               477             LD A,(DE):IF A=0 RET
3EC1 1B EB 36 20            478             DEC DE:EX DE,HL:LD (HL)," "
3EC5 06 FE                  479             LD B,254
3EC7 1A FE 0D 28 06         480 DELCR1 LD A,(DE):IF A=0DH JR DELCR2
3ECC 13 10 F8               481             INC DE:DJNZ DELCR1
3ECF 36 0D C9               482             LD (HL),0DH:RET
3ED2 ED 5B 5F 30 B7 ED 52   483 DELCR2 LD DE,(lineadr):SUB HL,DE
3ED9 7D 32 7E 30            484             LD A,L:LD (pointer),A
3EDD AF 32 2E 30            485             XOR A :LD (l_offs),A
3EE1 CD E0 39 C3 AA 33      486             CALL OFFSTR:JP TXTPRT
3EE7                        487 ;------------------------
3EE7                        488 MARK   :カット&ペースト
3EE7                        489            ;マーク&ジャンプ用 マーク
3EE7 CD 48 35 DA 80 36      490             CALL WRTLIN:JP C,MEMOVR
3EED 00 00 00               491             DS 3 ;for PATCH of MARK & JUMP
3EF0                        492                  ;そのうちに 付ますので しばらく お待ちを
3EF0 2A 5D 30 22 7A 30      493             LD HL,(lineno):LD (markbf),HL
3EF6 2A 32 30 25            494             LD HL,(xstart):DEC H
3EFA 3A 34 30 D6 08         495             LD A,(xsize):SUB 8
3EFF 85 6F CD 1E 20         496             ADD A,L:LD L,A:CALL #LOC
3F04 3E 2A C3 F4 1F         497             LD A,"*":JP #PRINT
3F09                        498 ;--------
3F09                        499 GETXT  ;テキスト取り込み
3F09 CD 48 35 DA 80 36      500             CALL WRTLIN:JP C,MEMOVR
3F0F CD 16 3F DA C4 1F      501             CALL GETXT1:JP C,#BELL
3F15 C9                     502             RET
3F16 2A 7A 30 23            503 GETXT1 LD HL,(markbf):INC HL
3F1A 7C B5 37 C8            504             LD A,H:OR L:SCF:RET Z
3F1E 2B CD 04 35 EB         505             DEC HL:CALL LINADR:EX DE,HL
3F23 2A 5D 30 23 CD 04 35   506             LD HL,(lineno):INC HL:CALL LINADR
3F2A B7 ED 52 44 4D D8      507             SUB HL,DE:LD BC,HL:RET C
3F30 37 C8                  508             SCF:RET Z
3F32 C5 D5                  509             PUSH BC:PUSH DE
3F34 2A 1E 30 ED 5B 1C 30   510              LD HL,(copybf):LD DE,(cutbf)
3F3B B7 ED 52 2B B7 ED 42   511              SUB HL,DE:DEC HL:SUB HL,BC
3F42 E1 C1                  512             POP HL:POP BC
3F44 C5 D5                  513             PUSH BC:PUSH DE
3F46 CD 97 1F               514              CALL #POKE@
3F49 E1 C1                  515             POP HL:POP BC
3F4B 09 AF CD 9A 1F         516             ADD HL,BC:XOR A:CALL #POKE
3F50 2A 29 30 ED 5B 1C 30   517             LD HL,(delbf):LD DE,(cutbf)
3F57 B7 ED 52 C0            518             SUB HL,DE:RET NZ
3F5B 21 00 00 22 78 30 C9   519             LD HL,0:LD (delpnt),HL:RET
3F62                        520 ;--------
3F62                        521 CUTXT  ;テキスト取り込み&テキストクリア
3F62 CD 48 35 DA 80 36      522             CALL WRTLIN:JP C,MEMOVR
3F68 CD 16 3F DA C4 1F      523             CALL GETXT1:JP C,#BELL
3F6E 2A 7A 30 CD 04 35      524             LD HL,(markbf):CALL LINADR
3F74 E5                     525             PUSH HL
3F75 2A 5D 30 23            526              LD HL,(lineno):INC HL
3F79 CD 04 35 EB            527              CALL LINADR:EX DE,HL
3F7D 2A 65 30               528              LD HL,(endadr)
3F80 AF ED 52 44 4D         529              XOR A:SBC HL,DE:LD BC,HL
3F85 E1                     530             POP HL
3F86 20 03 77 18 04         531             IF Z THEN LD (HL),A:JR CUTXT1
3F8B EB 03 ED B0            532             EX DE,HL:INC BC:LDIR
3F8F CD EC 34               533 CUTXT1 CALL ENDCHK
3F92 2A 7A 30 22 5D 30      534             LD HL,(markbf):LD (lineno),HL
3F98 CD 3B 34 C3 AA 33      535             CALL LINSET:JP TXTPRT
3F9E                        536 ;--------
3F9E                        537 PUTXT  ;テキスト出力
3F9E CD 48 35 DA 80 36      538             CALL WRTLIN:JP C,MEMOVR
3FA4 2A 1C 30 CD 94 1F      539             LD HL,(cutbf):CALL #PEEK
3FAA B7 C8                  540             IF A=0 RET
3FAC 01 00 00               541             LD BC,0
3FAF CD 94 1F B7 28 04      542 PUTXT1 CALL #PEEK:IF A=0 JR PUTXT2
3FB5 23 03 18 F6            543             INC HL:INC BC:JR PUTXT1
3FB9 CD 06 36 03            544 PUTXT2 CALL LEFBYT:INC BC
3FBD B7 ED 42 DA C4 1F      545             SUB HL,BC:JP C,#BELL
3FC3 C5                     546             PUSH BC
3FC4 2A 5D 30 CD 04 35      547              LD HL,(lineno):CALL LINADR
3FCA C1                     548             POP BC
3FCB 7E B7 20 10            549             LD A,(HL):IF A<>0 JR PUTXT4
3FCF ED 5B 1C 30 CD 91 1F   550 PUTXT3 LD DE,(cutbf):CALL #PEEK@:CALL ENDCHK
3FD6 CD EC 34
3FD9 CD 3B 34 C3 AA 33      551             CALL LINSET:JP TXTPRT
3FDF 0B                     552 PUTXT4 DEC BC
3FE0 E5 C5                  553             PUSH HL:PUSH BC
3FE2 50 59 44 4D 2A 65 30   554              LD DE,BC:LD BC,HL:LD HL,(endadr)
3FE9 E5 E5                  555              PUSH HL:PUSH HL
3FEB 19 EB                  556                ADD HL,DE:EX DE,HL
3FED E1                     557               POP HL
3FEE B7 ED 42 44 4D 03      558               SUB HL,BC:LD BC,HL:INC BC
3FF4 E1 ED B8               559              POP HL:LDDR
3FF7 C1 E1                  560             POP BC:POP HL
3FF9 18 D4                  561             JR PUTXT3
3FFB                        562 ;------------------------
3FFB                        563 JPWIN  ;ダイレクトに各ウィンドウにジャンプ
3FFB CD 48 35 DA 80 36      564             CALL WRTLIN:JP C,MEMOVR
4001 CD 2C 33               565             CALL TXTPUSH
4004 2A 2F 30 2E 00 CD 1E   566             LD HL,(scrnsiz):LD L,0:CALL #LOC
400B 20
400C 11 7A 31 CD E5 1F CD   567             LD DE,wjumpm:CALL #MSX:CALL INKEYSUB
4013 A9 36
4015 D6 30 DA 10 38         568             SUB "0":JP C,PDATA
401A FE 08 D2 10 38         569             IF A>=8 JP   PDATA
401F 21 FF FF 22 7A 30      570             LD HL,0-1:LD (markbf),HL
4025 32 31 30               571             LD (windno),A
4028 CD 84 32 CD EC 34      572             CALL WINDPOP:CALL ENDCHK
402E CD D8 33 C3 8A 33      573             CALL PRWSET:JP SCRNINIT+3
4034                        574 ;--------
4034                        575 WINMX  ;ウィンドウを最大にする
4034 CD 48 35 DA 80 36      576             CALL WRTLIN:JP C,MEMOVR
403A 3A 75 30 B7 20 10      577             LD A,(windmxf):IF A<>0 JR WINMX1
4040 3C 32 75 30            578             INC A:LD (windmxf),A
4044 21 50 1A 22 34 30      579             LD HL,01A50H:LD (xsize),HL
404A CD CF 32 C3 97 33      580             CALL WINDCHK:JP WTXTPRT
4050 AF 32 75 30            581 WINMX1 XOR A:LD (windmxf),A
4054 CD 2C 33 C3 52 33      582             CALL TXTPUSH:JP SCRNINIT_c
405A                        583 ;--------
405A                        584 WINMV  ;ウィンドウ  移動
405A 3A 70 30               585             LD A,(getlmode)
405D B7 C4 48 35            586             OR A:CALL NZ,WRTLIN
4061 DA 80 36               587             JP C,MEMOVR
4064 2A 32 30 25 2D         588 WINMV1 LD HL,(xstart):DEC H:DEC L
4069 CD 1E 20 2A 32 30      589             CALL #LOC:LD HL,(xstart)
406F CD C9 40 38 08         590             CALL WINVEC:JR C,WINMV2
4074 22 32 30 CD CF 32      591             LD (xstart),HL:CALL WINDCHK
407A 18 E8                  592             JR WINMV1
407C                        593             ;
407C 3A 70 30 B7 28 06      594 WINMV2 LD A,(getlmode):IF A=0 JR WINMV3
4082 CD 1A 33 C3 52 33      595             CALL WINDPUSH:JP SCRNINIT_c
4088                        596             ;
4088 3A 5B 30 32 5A 30      597 WINMV3 LD A,(comlin):LD (comtop),A
408E 2A 32 30 22 56 30      598             LD HL,(xstart):LD (comwind),HL
4094 2A 34 30 22 58 30      599             LD HL,(xsize) :LD (comwind+2),HL
409A C3 52 33               600             JP SCRNINIT_c
409D                        601 ;--------
409D                        602 WINSZ  ;ウィンドウ  大きさ指定
409D 3A 70 30               603             LD A,(getlmode)
40A0 B7 C4 48 35            604             OR A:CALL NZ,WRTLIN
40A4 DA 80 36               605             JP C,MEMOVR
40A7 2A 32 30 ED 5B 34 30   606 WINSZ1 LD HL,(xstart):LD DE,(xsize):ADD HL,DE
40AE 19
40AF CD 1E 20 2A 34 30      607             CALL #LOC:LD HL,(xsize)
40B5 CD C9 40 38 C2         608             CALL WINVEC:JR C,WINMV2
40BA 22 34 30 CD CF 32      609             LD (xsize),HL:CALL WINDCHK
40C0 2A 32 30 ED 4B 34 30   610             LD HL,(xstart):LD BC,(xsize)
40C7 18 DE                  611             JR WINSZ1
40C9                        612 ;
40C9 CD A9 36 CD F0 35      613 WINVEC CALL INKEYSUB:CALL UPPER
40CF FE 5A 28 58            614             IF A="Z" JR WINVEC1
40D3 FE 58 28 55            615             IF A="X" JR WINVEC2
40D7 FE 43 28 4E            616             IF A="C" JR WINVEC3
40DB FE 41 28 4F            617             IF A="A" JR WINVEC4
40DF FE 53 28 E6            618             IF A="S" JR WINVEC
40E3 FE 44 28 49            619             IF A="D" JR WINVEC6
40E7 FE 51 28 4A            620             IF A="Q" JR WINVEC7
40EB FE 57 28 47            621             IF A="W" JR WINVEC8
40EF FE 45 28 40            622             IF A="E" JR WINVEC9
40F3 FE 1F 28 35            623             IF A=1FH JR WINVEC2
40F7 FE 1D 28 33            624             IF A=1DH JR WINVEC4
40FB FE 1C 28 31            625             IF A=1CH JR WINVEC6
40FF FE 1E 28 33            626             IF A=1EH JR WINVEC8
4103 FE 31 28 24            627             IF A="1" JR WINVEC1
4107 FE 32 28 21            628             IF A="2" JR WINVEC2
410B FE 33 28 1A            629             IF A="3" JR WINVEC3
410F FE 34 28 1B            630             IF A="4" JR WINVEC4
4113 FE 35 28 B2            631             IF A="5" JR WINVEC
4117 FE 36 28 15            632             IF A="6" JR WINVEC6
411B FE 37 28 16            633             IF A="7" JR WINVEC7
411F FE 38 28 13            634             IF A="8" JR WINVEC8
4123 FE 39 28 0C            635             IF A="9" JR WINVEC9
4127 37 C9                  636             SCF:RET
4129 2C 3E                  637 WINVEC3 INC L:DB 3EH
412B 2D                     638 WINVEC1 DEC L
412C 24 3E                  639 WINVEC2 INC H:DB 3EH
412E 2D 3E                  640 WINVEC4 DEC L:DB 3EH
4130 2C B7 C9               641 WINVEC6 INC L:OR A:RET
4133 2C 3E                  642 WINVEC9 INC L:DB 3EH
4135 2D                     643 WINVEC7 DEC L
4136 25 B7 C9               644 WINVEC8 DEC H:OR A:RET
4139                        645 ;------------------------
4139                        646 PAGE3   ;LABEL for page3

0000                          1 ;-----------------------------
0000                          2 ;    WINER            page 3
0000                          3 ;       < screen editer >
0000                          4 ;    Copyright in 1988 by 環
0000                          5 ;-----------------------------
0000                          6
0000                          7             OFFSET  makadr-program
```

▶私はOh！Xを手にした翌日に入院してしまいました。原因は無理がたたっての肝臓病です。私が留守の間，女房はきっとX68000でゲームをしまくっていることでしょう。ああ，キーボードに触りたい。皆さん，仕事も大事だけど，なにより健康ですよ，健康が第一なのです。
佐藤　啓市（33）北海道

```
0000                       8
0000                       9 ;          ---------------
3282                      10            ORG     TO_COM
3282 39 41                11            DW      COMLVL
3284                      12 ;          ---------------
3284                      13
4139                      14            ORG     PAGE3
4139                      15
4139                      16 ;*************************
4139                      17 COMLVL  ; コマンド入力  コールド
4139                      18            ; des ALL
4139                      19
4139 CD 78 41             20            CALL COMLVLC
413C                      21
413C                      22 COMLVLH ; コマンド入力  ホット
413C                      23            ; des ALL
413C                      24
413C ED 7B 0C 30 CD 7A 42 25            LD SP,(stack):CALL PCOMLIN
4143 CD 3B 35 CD F7 41    26 COMLVL1 CALL LBFCLA:CALL COMRED
4149 CD DD 41 CD A9 38    27            CALL CONCUR:CALL GETL
414F CD D1 35 CD 10 42    28            CALL SET0DH:CALL COMWRT
4155 3A 80 30 FE 0D 20 0C 29            LD A,(chrbf):IF A<>0DH JR COMLVL2
415C                      30            :
415C CD 49 43             31            CALL CURD_C
415F ED 5B 10 30 CD 8B 45 32            LD DE,(linebf):CALL COMMAND_GO
4166 18 D4                33            JR COMLVLH
4168                      34            ;
4168 CD D6 42 20 D6       35 COMLVL2 CALL CTRLSEL_C:JR NZ,COMLVL1
416D CD EC 42 18 CA       36            CALL CTRLJP_C:JR COMLVLH
4172                      37
4172                      38            ;
4172 DC 99 36 C3 39 41    39 COM_c   CALL C,ERROR:JP COMLVL
4178                      40            ;
4178 2A 56 30 22 32 30    41 COMLVLC LD HL,(comwind)  :LD (xstart),HL
417E 2A 58 30 22 34 30    42            LD HL,(comwind+2):LD (xsize),HL
4184 CD CF 32             43 COMLVLC1 CALL WINDCHK
4187 2A 32 30 ED 4B 34 30 44            LD HL,(xstart):LD BC,(xsize)
418E CD 00 4C             45            CALL OWINDOW
4191 AF 32 70 30          46            XOR A:LD (getlmode),A
4195 32 7E 30 32 2E 30    47            LD (pointer),A:LD (l_offs),A
419B CD B5 37 CD 7A 42    48            CALL PCURDAT:CALL PCOMLIN
41A1 CD 10 38 C3 37 42    49            CALL PDATA:JP COMPRT
41A7                      50
41A7                      51 ;------------------------
41A7                      52 COMADR  ; コマンドアドレス
41A7                      53            ; in A=line No. /out HL=command buff adress
41A7                      54
41A7 D5                   55            PUSH DE
41A8 57 1E 00 CB 3A CB 1B 56             LD D,A:LD E,0:SRL D:RR E
41AF 2A 26 30 19          57             LD HL,(combf):ADD HL,DE
41B3 D1 C9                58            POP DE:RET
41B5                      59
41B5                      60 COMDEC  ; コマンドNoデクリメント
41B5 3D 3D                61            DEC A:DEC A
41B7                      62 COMINC  ; コマンドNoインクリメント
41B7 C5                   63            PUSH BC
41B8 3C 4F                64             INC A:LD C,A
41BA 3A 28 30 A1          65             LD A,(commsk):AND C
41BE C1 C9                66            POP BC:RET
41C0                      67
41C0                      68 COMEINC ; コマンドエンドマーク  インクリメント
41C0 3A 5B 30 CD A7 41    69            LD A,(comlin):CALL COMADR
41C6 E5                   70            PUSH HL
41C7 F5                   71             PUSH AF
41C8 CD 94 1F B7 20 0C    72              CALL #PEEK:IF A<>0 JR COMEINC1
41CE F1 F5                73             POP AF:PUSH AF
41D0 CD B7 41 CD A7 41    74              CALL COMINC:CALL COMADR
41D6 AF CD 9A 1F          75              XOR A:CALL #POKE
41DA F1                   76 COMEINC1 POP AF
41DB E1 C9                77            POP HL:RET
41DD                      78
41DD                      79 CONCUR  ; コマンドラインポジションセット
41DD                      80            ; des AF,HL,C
41DD                      81
41DD 3A 5A 30 67          82            LD A,(comtop):LD H,A
41E1 3A 5B 30 6F          83            LD A,(comlin):LD L,A
41E5 94 30 06             84            SUB H:JR NC,CONCUR1
41E8 3A 28 30 94 85 3C    85            LD A,(commsk):SUB H:ADD A,L:INC A
41EE 2A 32 30 84 67       86 CONCUR1 LD HL,(xstart):ADD A,H:LD H,A
41F3 22 2C 30 C9          87            LD (l_curx),HL:RET
41F7                      88
41F7                      89 COMRED  ; コマンドバッファから読み込む
41F7                      90            ; in HL=command buff adress
41F7                      91            ; des AF,HL,DE,C
41F7                      92
41F7 3A 5B 30 CD A7 41    93            LD A,(comlin):CALL COMADR
41FD                      94            ;
41FD ED 5B 10 30 0E 7F    95 COMRED1 LD DE,(linebf):LD C,127
4203 CD 94 1F FE 20 D8    96 COMRED2 CALL #PEEK:IF A<20H RET
4209 12 13 23             97            LD (DE),A:INC DE:INC HL
420C 0D 20 F4             98            IF DEC(C)<>0 JR COMRED2
420F C9                   99            RET
4210                     100
4210                     101 COMWRT  ; コマンドバッファに  書き込む
4210                     102            ; in HL=command buff adress
4210                     103            ; des AF,HL,DE,C
4210                     104
4210 CD C0 41            105            CALL COMEINC
4213 3A 7F 30 B7 20 07   106            LD A,(inpchk):IF A<>0 JR COMWRT1
4219 CD 94 1F B7 28 13   107            CALL #PEEK:IF A=0 JR COMWRT3
421F C9                  108            RET
4220 ED 5B 10 30 0E 7F   109 COMWRT1 LD DE,(linebf):LD C,127
4226 1A CD 9A 1F         110 COMWRT2 LD A,(DE):CALL #POKE
422A 13 23 FE 0D C8      111            INC DE:INC HL:IF A=0DH RET
422F 0D 20 F4            112            IF DEC(C)<>0 JR COMWRT2
4232 3E 0D C3 9A 1F      113 COMWRT3 LD A,0DH:JP   #POKE
4237                     114
4237                     115 COMPRT  ; コマンドスクリーン表示
4237                     116            ; des AF,HL,DE,BC
4237                     117
4237 CD A9 34            118            CALL TABPRT
423A 3A 35 30 47         119            LD A,(ysize):LD B,A
423E 2A 32 30 22 2C 30   120            LD HL,(xstart):LD (l_curx),HL
4244 3A 5A 30            121            LD A,(comtop)
4247 F5                  122 COMPRT1 PUSH AF
4248 CD A7 41 CD 94 1F   123             CALL COMADR:CALL #PEEK
424E B7 28 18            124             IF A=0 JR COMPRT2
4251 CD FD 41 3E 0D 12   125             CALL COMRED1:LD A,0DH:LD (DE),A
4257 ED 5B 10 30 CD 03 4C 126            LD DE,(linebf):CALL LNPRINT
425E 21 2D 30 34         127             LD HL,l_cury:INC (HL)
4262 F1                  128            POP AF
4263 CD B7 41 10 DF      129            CALL COMINC:DJNZ COMPRT1
4268 C9                  130            RET
4269 E1                  131 COMPRT2  POP HL
426A 2A 10 30 36 0D      132            LD HL,(linebf):LD (HL),0DH
426F EB 21 2D 30         133            EX DE,HL:LD HL,l_cury
4273 CD 03 4C 34         134 COMPRT3 CALL LNPRINT:INC (HL)
4277 10 FA               135            DJNZ COMPRT3
4279 C9                  136            RET
427A                     137
427A                     138 PCOMLIN ; コマンドスクリーンのラインNo.表示
427A                     139            ; des AF,HL,DE
427A                     140
427A 2A 32 30 25         141            LD HL,(xstart):DEC H
427E 3A 34 30 D6 07      142            LD A,(xsize):SUB 7
4283 85 6F CD 1E 20      143            ADD A,L:LD L,A:CALL #LOC
4288 3A 5B 30 0E 00      144            LD A,(comlin):LD C,0
428D CD B5 41 CD A7 41   145 PCOMLIN1 CALL COMDEC:CALL COMADR
4293 08                  146             EX AF,AF'
4294 CD 94 1F B7 28 04   147             CALL #PEEK:IF A=0 JR PCOMLIN2
429A 08                  148             EX AF,AF'
429B 0C 18 EF            149             INC C:JR PCOMLIN1
429E 69 26 00 CD 39 36   150 PCOMLIN2 LD L,C:LD H,0:CALL SEISU
42A4 11 0A 31 C3 E5 1F   151            LD DE,namare:JP   #MSX
42AA                     152 ;
42AA                     153 COM#APT ;Accを(pointer)に合わせて
42AA                     154            ; ラインバッファに書き込む
42AA E5                  155            PUSH HL
42AB F5                  156             PUSH AF
42AC CD C4 39            157              CALL PINADR
42AF F1                  158             POP AF
42B0 77 23 36 0D         159             LD (HL),A:INC HL:LD (HL),0DH
42B4 21 7E 30 34         160             LD HL,pointer:INC (HL)
42B8 E1 C9               161            POP HL:RET
42BA                     162 ;
42BA                     163 COM#MSX ;DE以後 00Hまでを(pointer)に合わせて
42BA                     164            ; ラインバッファに書き込む
42BA 1A B7 C8            165            LD A,(DE):IF A=0 RET
42BD CD AA 42 13         166            CALL COM#APT:INC DE
42C1 18 F7               167            JR COM#MSX
42C3                     168 ;
42C3                     169 COM#PRT ;ラインバッファを
42C3                     170            ; コマンドテキストに書き込み表示する
42C3 3A 5B 30 CD A7 41 CD 171           LD A,(comlin):CALL COMADR:CALL COMWRT
42CA 10 42
42CC CD DD 41 ED 5B 10 30 172           CALL CONCUR:LD DE,(linebf):CALL LNPRINT
42D3 CD 03 4C
42D6                     173
42D6                     174 ;------------------------
42D6                     175 CTRLSEL_C ; コマンドスクリーンの
42D6                     176            ;   コントロールキーセレクト
42D6                     177            ; in A=ASCII CODE / Z=OK
42D6                     178
42D6 E5 C5               179            PUSH HL:PUSH BC
42D8 21 E3 42            180             LD HL,ctrlsel_c
42DB 01 09 00            181             LD BC,CTRLJP_C-ctrlsel_c
42DE ED B1               182             CPIR
42E0 C1 E1 C9            183            POP BC:POP HL:RET
42E3                     184            ;
42E3 05 0D 18 1B         185 ctrlsel_c DB 05H,0DH,18H,1BH
42E7 1E 1F 22 23 24      186             DB 1EH,1FH,22H,23H,24H
42EC                     187 CTRLJP_C  ; コマンドスクリーンの
42EC                     188            ;   コントロールキージャンプ
42EC                     189            ; in A=ASCII CODE
42EC                     190
42EC FE 05 28 1F         191            IF A=05H JR CURU_C
42F0 FE 0D 28 4E         192            IF A=0DH JR CR_C
42F4 FE 18 28 51         193            IF A=18H JR CURD_C
42F8 FE 1B 28 34         194            IF A=1BH JR COMBRK
42FC FE 1E 28 0F         195            IF A=1EH JR CURU_C
4300 FE 1F 28 45         196            IF A=1FH JR CURD_C
4304 FE 22 CA 83 43      197            IF A=22H JP WMX_C
4309 CD 09 39 C3 78 41   198            CALL CTRLJP:JP COMLVLC
430F                     199
430F                     200 CURU_C  ; コマンドスクリーンの
430F                     201            ;   カーソルアップ
430F                     202
430F 3A 5B 30 47         203            LD A,(comlin):LD B,A
4313 CD B5 41 CD A7 41   204            CALL COMDEC:CALL COMADR
4319 08                  205            EX AF,AF'
431A CD 94 1F B7 C8      206            CALL #PEEK:IF A=0 RET
431F 08 32 5B 30         207            EX AF,AF':LD (comlin),A
4323 08                  208            EX AF,AF'
4324 3A 5A 30 B8 C0      209            LD A,(comtop):IF A<>B RET
4329 08 32 5A 30         210            EX AF,AF':LD (comtop),A
432D C3 37 42            211            JP COMPRT
4330                     212
4330                     213 COMBRK  ; コマンドラインのブレイク
4330                     214
4330 AF 32 7E 30 32 2E 30 215           XOR A:LD (pointer),A:LD (l_offs),A
4337 3A 5B 30 CD A7 41   216            LD A,(comlin):CALL COMADR
433D 3E 0D C3 9A 1F      217            LD A,0DH:JP #POKE
4342                     218
4342                     219 CR_C    ; コマンドスクリーンのCR
4342                     220
4342 AF 32 7E 30 32 2E 30 221           XOR A:LD (pointer),A:LD (l_offs),A
4349                     222
4349                     223 CURD_C  ; コマンドスクリーンの
4349                     224            ;   カーソルダウン
4349                     225
4349 3A 5B 30 CD B7 41   226            LD A,(comlin):CALL COMINC
434F 32 5B 30 CD C0 41   227            LD (comlin),A:CALL COMEINC
4355 3A 5A 30 CD A7 41   228            LD A,(comtop):CALL COMADR
435B 4F CD 94 1F B7 20 0A 229           LD C,A:CALL #PEEK:IF A<>0 JR CURD_C1
4362 79 CD B7 41         230            LD A,C:CALL COMINC
4366 32 5A 30 C3 37 42   231            LD (comtop),A:JP COMPRT
436C 3A 35 30 47         232 CURD_C1 LD A,(ysize):LD B,A
4370 3A 5B 30            233            LD A,(comlin)
4373 05 28 07            234            IF DEC(B)=0 JR CURD_C3
4376 CD B5 41 B9 C8      235 CURD_C2 CALL COMDEC:IF A=C RET
437B 10 F9               236            DJNZ CURD_C2
437D 32 5A 30 C3 37 42   237 CURD_C3 LD (comtop),A:JP COMPRT
4383                     238
4383                     239 WMX_C   ;コマンドウィンドウを最大にする
4383                     240
4383 3A 75 30 B7 28 0A   241            LD A,(windmxf):IF A=0 JR WMX_C1
4389 AF 32 75 30         242            XOR A:LD (windmxf),A
438D CD 52 33 C3 78 41   243            CALL SCRNINIT_c:JP COMLVLC
4393 3C 32 75 30         244 WMX_C1  INC A:LD (windmxf),A
4397 21 50 1A 22 34 30 C3 245           LD HL,01A50H:LD (xsize),HL:JP COMLVLC1
439E 84 41
43A0                     246
43A0                     247 ;------------------------
43A0                     248 SYSRET  ; 呼び出したシステムに帰る
43A0                     249
43A0 ED 7B 0C 30         250            LD SP,(stack)
43A4 3E 0C C3 F4 1F      251            LD A,0CH:JP #PRINT
43A9                     252 ;
43A9                     253 TABSW   ; タブ表示切換え
43A9                     254
43A9 3A 6B 30 EE 01 32 6B 255           LD A,(tabsw):XOR 1:LD (tabsw),A
43B0 30
43B1 CD 52 33 C3 78 41   256            CALL SCRNINIT_c:JP COMLVLC
43B7                     257 ;
43B7                     258 TXTSEL  ; テキスト選択
43B7                     259
43B7 1A CD F0 35         260            LD A,(DE):CALL UPPER
43BB D6 41 D8            261            SUB "A":RET C
43BE FE 04 D0            262            IF A>=4 RET
43C1 4F 08 79            263            LD C,A:EX AF,AF':LD A,C
43C4 CD CC 34 DA 96 44   264            CALL TEXTDAT:JP C,TDATPRT2
43CA 08 32 5C 30         265            EX AF,AF':LD (nowtxt),A
43CE 21 00 00 22 5D 30   266            LD HL,0:LD  (lineno),HL
43D4 CD 2C 33 CD 84 32   267            CALL TXTPUSH:CALL WINDPOP
43DA CD EC 34 CD 87 33   268            CALL ENDCHK:CALL SCRNINIT
43E0 C3 78 41            269            JP COMLVLC
43E3                     270 ;
43E3                     271 TDATPRT ; テキストデータ表示
43E3                     272
43E3 1A 13               273            LD A,(DE):INC DE
43E5 FE 2F 20 32         274            IF A<>"/" JR TDATPRT0
43E9 1A 13 CD F0 35      275            LD A,(DE):INC DE:CALL UPPER
43EE D6 41 D8            276            SUB "A" RET C
43F1 FE 04 D0            277            IF A>=4 RET
43F4 87 DD 21 12 30      278            ADD A,A:LD IX,txtst
43F9 4F 06 00 DD 09      279            LD C,A:LD B,0:ADD IX,BC
43FE DD 4E 02 DD 46 03   280            LD C,(IX+2):LD B,(IX+3)
4404 13 CD B2 1F         281            INC DE:CALL #HLHEX
4408 E5 B7 ED 42 E1      282            PUSH HL:SUB HL,BC:POP HL
440D 38 03 C2 C4 1F      283            IF NC THEN JP NZ,#BELL
4412 DD 75 00 DD 74 01   284            LD (IX+0),L:LD (IX+1),H
4418 CD 10 34            285            CALL TAREAI
```


▶ついに X68000を買った。CZ-600C と CZ-800C はよく似合う。そして，ED-BETA にも似合う。預金通帳はいつの間にか3桁の世界である。　　奥田　健児 (18) 神奈川県

```
441B                    286                 ;
441B 01 00 03           287 TDATPRT0 LD BC,0300H
441E C5                 288                 PUSH BC
441F CD 26 44           289                  CALL TDATPRT1
4422 C1 0C              290                 POP BC:INC C
4424 10 F8              291                 DJNZ TDATPRT0+3
4426 3A 5C 30 91 3E 20  292 TDATPRT1 LD A,(nowtxt):SUB C:LD A," "
442C 20 02 3E 2A        293                 IF Z THEN LD A,"*"
4430 CD AA 42           294                          CALL COM#APT
4433 3E 2F CD AA 42     295                 LD A,"/":CALL COM#APT
4438 79 08              296                 LD A,C:EX AF,AF'
443A 79 C6 41 CD AA 42  297                 LD A,C:ADD A,"A":CALL COM#APT
4440 3E 3A CD AA 42     298                 LD A,":":CALL COM#APT
4445 79 CD CC 34 38 4B  299                 LD A,C:CALL TEXTDAT:JR C,TDATPRT2
444B D5 E5 C5           300                 PUSH DE:PUSH HL:PUSH BC
444E CD 9E 44 1B        301                   CALL TDATPRT3:DEC DE
4452 3E 2D CD AA 42     302                   LD A,"-":CALL COM#APT
4457 EB CD 9E 44        303                   EX DE,HL:CALL TDATPRT3
445B 3E 3A CD AA 42     304                   LD A,":":CALL COM#APT
4460 60 69 CD 9E 44     305                   LD HL,BC:CALL TDATPRT3
4465 3E 20 CD AA 42     306                   LD A," ":CALL COM#APT
446A 3E 5B CD AA 42     307                   LD A,"[":CALL COM#APT
446F C1 E1              308                  POP BC:POP HL
4471 AF ED B1 EB        309                  XOR A:CPIR:EX DE,HL
4475 E1                 310                 POP HL
4476 60 69 CD 9E 44     311                 LD HL,BC:CALL TDATPRT3
447B 3E 5D CD AA 42     312                 LD A,"]":CALL COM#APT
4480 08 CD B0 44        313                 EX AF,AF':CALL FNADR
4484 CD 94 1F 23        314 TDATPRT11 CALL #PEEK:INC HL
4488 B7 28 05           315                 IF A=0 JR TDATPRT12
448B CD AA 42 18 F4     316                 CALL COM#APT:JR TDATPRT11
4490 CD C3 42 C3 42 43  317 TDATPRT12 CALL COM#PRT:JP   CR_C
4496                    318                 ;
4496 11 98 31 CD BA 42  319 TDATPRT2 LD DE,nousem:CALL COM#MSX
449C 18 F2              320                 JR TDATPRT12
449E                    321                 ;
449E D5                 322 TDATPRT3 PUSH DE
449F EB CD C4 39        323                  EX DE,HL:CALL PINADR
44A3 EB CD BC 44        324                  EX DE,HL:CALL HLASC
44A7 21 7E 30 34        325                  LD HL,pointer:INC (HL)
44AB 34 34 34           326                  INC (HL):INC (HL):INC (HL)
44AE D1 C9              327                 POP DE:RET
44B0                    328 ;
44B0                    329 FNADR   ;ファイルネームバッファアドレス算出
44B0                    330                 ; in  Acc=text No.
44B0                    331                 ; out HL=buff adress
44B0                    332                 ; des DE
44B0 3C 11 20 00 2A 20 30  333              INC A:LD DE,20H:LD HL,(txtwbf)
44B7 19 3D 20 FC        334 FNADR1  ADD HL,DE:IF DEC(A)<>0 JR FNADR1
44BB C9                 335                 RET
44BC                    336
44BC                    337
44BC                    338 HLASC   ; １６進数を ASCII CODE に変換
44BC 7C CD C1 44        339                 LD A,H:CALL HXASC
44C0 7D                 340                 LD A,L
44C1 F5                 341 HXASC   PUSH AF
44C2 0F 0F 0F 0F        342                  RRCA:RRCA:RRCA:RRCA
44C6 CD CA 44           343                  CALL HXASC1
44C9 F1                 344                 POP AF
44CA E6 0F F6 30        345 HXASC1  AND 0FH:OR 30H
44CE FE 3A 38 02 C6 07  346                 IF A>=3AH THEN ADD A,7
44D4 12 13 C9           347                 LD (DE),A:INC DE:RET
44D7                    348
44D7                    349 TXTCLA  ; テキストクリア
44D7                    350
44D7 11 6C 30 3A 5C 30  351                 LD DE,txtchr:LD A,(nowtxt)
44DD 83 5F 30 01 14     352                 ADD A,E:LD E,A:IF C THEN INC D
44E2 2A 67 30 7E        353                 LD HL,(ntxtst):LD A,(HL)
44E6 B7 C8              354                 IF A=0 RET
44E8 12 36 00           355                 LD (DE),A:LD (HL),0
44EB 3A 5C 30 CD B0 44 CD  356              LD A,(nowtxt):CALL FNADR:CALL #POKE
44F2 9A 1F
44F4 CD 52 33 C3 78 41  357                 CALL SCRNINIT_c:JP COMLVLC
44FA                    358
44FA                    359 TXTREM  ; テキスト復活
44FA                    360
44FA 3A 5C 30 11 6C 30  361                 LD A,(nowtxt):LD DE,txtchr
4500 83 5F 30 01 14     362                 ADD A,E:LD E,A:IF C THEN INC D
4505 2A 67 30           363                 LD HL,(ntxtst)
4508 7E B7 C0           364                 LD A,(HL):IF A<>0 RET
450B 1A B7 C8           365                 LD A,(DE):IF A=0 RET
450E 77 CD EC 34        366                 LD (HL),A:CALL ENDCHK
4512 CD 52 33 C3 78 41  367                 CALL SCRNINIT_c:JP COMLVLC
4518                    368 ;
4518                    369 WDATPRT ; ウィンドウデータ表示
4518                    370
4518 0E 00 06 04        371                 LD C,0:LD B,4
451C CD 23 45           372                 CALL WDATPRT1
451F 0E 04 06 04        373                 LD C,4:LD B,4
4523 CD 2F 45 0C        374 WDATPRT1 CALL WDATPRT2:INC C
4527 10 FA              375                 DJNZ WDATPRT1
4529 CD C3 42 C3 42 43  376                 CALL COM#PRT:JP CR_C
452F 3A 31 30 91        377 WDATPRT2 LD A,(windno):SUB C
4533 3E 20 20 02 3E 2A  378                  LD A," ":IF Z THEN LD A,"*"
4539 CD AA 42           379                          CALL COM#APT
453C 79 C6 30           380                 LD A,C:ADD A,"0"
453F CD AA 42           381                          CALL COM#APT
4542 3E 2F CD AA 42     382                 LD A,"/":CALL COM#APT
4547 79 87 81           383                 LD A,C:ADD A,A:ADD A,C
454A 2A 20 30 85        384                 LD HL,(txtwbf):ADD A,L
454E 6F 30 01 24        385                 LD L,A:IF C THEN INC H
4552 CD 94 1F           386                 CALL #PEEK
4555 C6 41 CD AA 42     387                 ADD A,"A":CALL COM#APT
455A 3E 3A C3 AA 42     388                 LD A,":" :JP   COM#APT
455F                    389 ;
455F                    390 WINDSEL ; ウィンドウ選択
455F                    391
455F D6 30              392                 SUB "0"
4561 FE 08 D0           393                 IF A>=8 RET
4564 D5                 394                 PUSH DE
4565 F5                 395                  PUSH AF
4566 CD 2C 33           396                   CALL TXTPUSH
4569 F1                 397                  POP AF
456A 32 31 30 CD 84 32  398                  LD (windno),A:CALL WINDPOP
4570 CD EC 34 CD 8A 33  399                  CALL ENDCHK:CALL SCRNINIT+3
4576 CD D8 33 CD 78 41  400                  CALL PRWSET:CALL COMLVLC
457C D1 18 0C           401                 POP DE:JR  COMMAND_GO
457F                    402 ;
457F                    403 LINTOP  ;ラインNoとアドレスをトップにする
457F 21 00 00 22 5D 30  404                 LD HL,0:LD (lineno),HL
4585 2A 67 30 22 5F 30  405                 LD HL,(ntxtst):LD (lineadr),HL
458B                    406
458B                    407 ;-----------------------
458B                    408 COMMAND_GO ;コマンド実行
458B                    409
458B D5                 410                 PUSH DE
458C 3A 5B 30 CD B5 41  411                  LD A,(comlin):CALL COMDEC
4592 CD A7 41 CD FD 41  412                  CALL COMADR:CALL COMRED1
4598 3E 0D 12           413                  LD A,0DH:LD (DE),A
459B AF 13 12           414                  XOR A:INC DE:LD (DE),A
459E D1                 415                 POP DE
459F 1A                 416                 LD A,(DE)
45A0 13 CD F0 35        417                 INC DE:CALL UPPER
45A4 CD 32 36 30 B6     418                 CALL CHKNUM:JR NC,WINDSEL
45A9 FE 42 28 D2        419                 IF A="B" JR LINTOP
45AD FE 21 CA A0 43     420                 IF A="!" JP SYSRET
45B2 FE 25 CA A9 43     421                 IF A="%" JP TABSW
45B7 FE 26 CA D7 44     422                 IF A="&" JP TXTCLA
45BC FE 2F CA B7 43     423                 IF A="/" JP TXTSEL
45C1 FE 52 CA FA 44     424                 IF A="R" JP TXTREM
45C6 FE 4D CA E3 43     425                 IF A="M" JP TDATPRT
45CB FE 4E CA 18 45     426                 IF A="N" JP WDATPRT
45D0                    427
45D0                    428 PAGE4   ;for page 4 LABEL
45D0 C9                 429                 RET

0000                      1 ;-----------------------------
0000                      2 ;    WINER          page 4
0000                      3 ;       < screen editer >
0000                      4 ;    Copyright in 1988 by 環
0000                      5 ;-----------------------------
0000                      6
0000                      7                 OFFSET  makadr-program
0000                      8
0000                      9 ;       ------------
45D0                     10                 ORG     PAGE4
45D0                     11 ;       ------------
45D0                     12
45D0 FE 45 CA FE 45      13                 IF A="E" JP EDIT
45D5 FE 4C CA 6D 46      14                 IF A="L" JP LOAD
45DA FE 53 CA 48 47      15                 IF A="S" JP SAVE
45DF FE 46 CA 06 48      16                 IF A="F" JP SEARCH
45E4 FE 43 CA EB 48      17                 IF A="C" JP CHENGE
45E9 FE 44 CA 1D 4A      18                 IF A="D" JP DELTXT
45EE FE 58 CA 6F 4A      19                 IF A="X" JP SWPTXT
45F3 FE 50 CA 46 4B      20                 IF A="P" JP PRWSW
45F8 00 00 00 00 00      21                 DS  5           ; 拡張用
45FD C9                  22                 RET
45FE                     23
45FE                     24 ;-----------------------
45FE                     25 EDIT    ;スクリーンエディット
45FE                     26
45FE D5                  27                 PUSH DE
45FF CD 84 32 CD EC 34   28                  CALL WINDPOP:CALL ENDCHK
4605 D1                  29                 POP DE
4606 2A 5D 30 CD E9 35   30                 LD HL,(lineno):CALL SPSKIP
460C CD 32 36 D4 12 36   31                 CALL CHKNUM:CALL NC,GETNUM
4612 22 5D 30 CD 3B 34   32                 LD (lineno),HL:CALL LINSET
4618 CD 97 33            33                 CALL WTXTPRT
461B 21 FF FF 22 7A 30   34                 LD HL,0-1:LD (markbf),HL
4621                     35                 ;
4621 3E 01 32 70 30      36 EDIT1   LD A,1:LD (getlmode),A
4626 CD 26 35 CD 59 46   37                 CALL LINTRS:CALL EDCUR
462C CD 29 38 CD D1 37   38                 CALL PLEFDAT:CALL PLINDAT
4632 CD A9 38 2A 10 30   39                 CALL GETL:LD HL,(linebf)
4638 7E FE 1B 28 08      40                 LD A,(HL):IF A=1BH JR EDITEND
463D 3A 80 30 CD 09 39   41                 LD A,(chrbf) :CALL CTRLJP
4643 18 E1               42                 JR EDIT1+5
4645 CD 2C 33 3A 75 30   43 EDITEND CALL TXTPUSH:LD A,(windmxf)
464B B7 20 05 CD AA 33 18  44               IF A=0 THEN CALL TXTPRT  ELSE CALL SCRNINIT_c
4652 03 CD 52 33
4656 C3 78 41            45                 JP COMLVLC
4659                     46 ;
4659 ED 4B 61 30 2A 5D 30  47 EDCUR   LD BC,(dtopno):LD HL,(lineno)
4660 B7 ED 42 7D         48                 SUB HL,BC:LD A,L
4664 2A 32 30 84 67      49                 LD HL,(xstart):ADD A,H:LD H,A
4669 22 2C 30 C9         50                 LD (l_curx),HL:RET
466D                     51 ;-----------------------
466D                     52 LOAD    ; ファイルロード
466D CD B5 46 38 33      53                 CALL GETFILE:JR C,LOADER
4672 20 3B               54                             JR NZ,LOAD1
4674 3E 0C CD F4 1F      55                 LD A,0CH:CALL #PRINT
4679 CD 84 32 CD EC 34   56                 CALL WINDPOP:CALL ENDCHK
467F CD 06 36 ED 4B 72 1F  57               CALL LEFBYT:LD BC,(#SIZE)
4686 AF ED 42 38 1A      58                 XOR A:SBC HL,BC:JR C,LOADER
468B 11 A1 31 CD E5 1F CD  59               LD DE,loadm:CALL #MSX:CALL #FPRNT
4692 9D 1F
4694 2A 65 30 22 70 1F   60                 LD HL,(endadr):LD (#DTADR),HL
469A CD A6 1F 38 06      61                 CALL #RDD:JR C,LOADER
469F CD AF 46 C3 00 47   62                 CALL LOAD1:JP FNSET
46A5                     63                 ;
46A5 F5                  64 LOADER  PUSH AF
46A6 3E 0C CD F4 1F      65                  LD A,0CH:CALL #PRINT
46AB F1                  66                 POP AF
46AC CD 99 36            67                 CALL ERROR
46AF CD 52 33 C3 78 41   68 LOAD1   CALL SCRNINIT_c:JP COMLVLC
46B5                     69 ;
46B5                     70 GETFILE ;フィルネーム入力
46B5 CD EC 34 CD E9 35   71                 CALL ENDCHK:CALL SPSKIP
46BB 3E 04 CD A3 1F      72                 LD A,4:CALL #FILE
46C0 CD 09 20 C8         73                 CALL #ROPEN:RET Z
46C4 FE 08 37 C0         74                 CP 8:SCF:RET NZ
46C8 CD D7 46 D8 C0      75 GETFILE1 CALL GETFILE2:RET C:RET NZ
46CD CD 09 20 C8         76                 CALL #ROPEN:RET Z
46D1 FE 08 28 F3         77                 IF A=8 JR GETFILE1
46D5 37 C9               78                 SCF:RET
46D7                     79
46D7                     80 GETFILE2;ディレクトリー表示してフィルネーム入力
46D7 3E 0C CD F4 1F      81                 LD A,0CH:CALL #PRINT
46DC CD 06 20 D8         82                 CALL #DIR:RET C
46E0 11 5E 31 CD E5 1F   83                 LD DE,dirm:CALL #MSX
46E6 ED 5B 76 1F CD D3 1F  84               LD DE,(#KBFAD):CALL #GETL
46ED 1A FE 1B 20 02 B7 C9  85               LD A,(DE):IF A=1BH THEN OR A:RET
46F4 13 13               86                 INC DE:INC DE:
46F6 13 13 13            87                 INC DE:INC DE:INC DE
46F9 3E 04 CD A3 1F      88                 LD A,4:CALL #FILE
46FE AF C9               89                 XOR A:RET
4700                     90 ;
4700                     91 FNSET   ;ファイルネームを
4700                     92                 ;コマンドウィンドウに表示して
4700                     93                 ;ファイルネームバッファに入れる
4700 3A 5C 30 CD B0 44   94                 LD A,(nowtxt):CALL FNADR
4706 3E 20 CD 40 47      95                 LD A," ":CALL FNSET3
470B 3E 2A CD AA 42      96                 LD A,"*":CALL COM#APT
4710 3A 5D 1F CD 40 47   97                 LD A,(#DSK):CALL FNSET3
4716 3E 3A CD 40 47      98                 LD A,":"   :CALL FNSET3
471B ED 5B 74 1F         99                 LD DE,(#IBFAD)
471F 06 0D CD 34 47     100                 LD B,13:CALL FNSET2
4724 3E 2E CD 40 47     101                 LD A,"."   :CALL FNSET3
4729 06 03 CD 34 47     102                 LD B,3:CALL FNSET2
472E CD C3 42 C3 42 43  103                 CALL COM#PRT:JP CR_C
4734 13 1A CD AA 42     104 FNSET2  INC DE:LD A,(DE):CALL COM#APT
4739 CD 9A 1F 23        105                 CALL #POKE:INC HL
473D 10 F5 C9           106                 DJNZ FNSET2:RET
4740 CD AA 42 CD 9A 1F  107 FNSET3  CALL COM#APT:CALL #POKE
4746 23 C9              108                 INC HL:RET
4748                    109 ;-----------------------
4748                    110 SAVE    ; ファイルセーブ
4748 CD EC 34 CD E9 35  111                 CALL ENDCHK:CALL SPSKIP
474E FE 0D 28 62        112                 IF A=0DH JR SAVE5
4752 D5 3E 04 CD A3 1F D1  113 SAVE1   PUSH DE:LD A,4:CALL #FILE:POP DE
4759 13 1A              114                 INC DE:LD A,(DE)
475B FE 3A 20 0F        115                 IF A<>":" JR SAVE4
475F 13 1A              116                 INC DE:LD A,(DE)
4761 FE 20 30 09        117                 IF A>=20H JR SAVE4
4765 CD D7 46           118 SAVE3   CALL GETFILE2
4768 DA A5 46           119                 JP C,LOADER
476B C2 AF 46           120                 JP NZ,LOAD1
476E 3E 0C CD F4 1F     121 SAVE4   LD A,0CH:CALL #PRINT
4773 11 BE 31 CD E5 1F CD  122              LD DE,savem:CALL #MSX:CALL #FPRNT
477A 9D 1F
477C CD 84 32 CD EC 34  123                 CALL WINDPOP:CALL ENDCHK
4782 ED 5B 67 30 2A 65 30  124              LD DE,(ntxtst):LD HL,(endadr):INC HL
4789 23
478A AF ED 52 DA A5 46  125                 XOR A:SBC HL,DE:JP C,LOADER
4790 22 72 1F           126                 LD (#SIZE),HL
4793 21 00 00 22 70 1F 22  127              LD HL,0:LD (#DTADR),HL:LD (#EXADR),HL
479A 6E 1F
479C CD AF 1F DA A5 46  128                 CALL #WOPEN:JP C,LOADER
47A2 2A 67 30 22 70 1F  129                 LD HL,(ntxtst):LD (#DTADR),HL
47A8 CD AC 1F DA A5 46  130                 CALL #WRD:JP C,LOADER
47AE CD AF 46 C3 00 47  131                 CALL LOAD1:JP FNSET
```

▶最近の日本ファルコムはX1ユーザーにとても冷たい。　保延　陽一（14）東京都

```
47B4                       132
47B4 3A 5C 30 CD B0 44     133 SAVE5   LD A,(nowtxt):CALL FNADR
47BA CD 94 1F B7 28 A5     134             CALL #PEEK:IF A=0 JR SAVE3
47C0 3E 20 CD AA 42        135             LD A," ":CALL COM#APT
47C5 CD 94 1F 23           136 SAVE7   CALL #PEEK:INC HL
47C9 B7 28 05 CD AA 42 18  137             IF A<>0 THEN CALL COM#APT:JR SAVE7
47D0 F4
47D1 3E 3A CD AA 42        138             LD A,":":CALL COM#APT
47D6 11 AE 31 CD BA 42     139             LD DE,save?:CALL COM#MSX
47DC CD E0 39 CD C3 42     140             CALL OFFSTR:CALL COM#PRT
47E2 CD A0 37 CD A9 36     141             CALL INKEYCUR:CALL INKEYSUB
47E8 FE 1B CA 42 43        142             IF A=1BH JP CR_C
47ED FE 0D 28 06 CD 42 43  143             IF A<>0DH THEN CALL CR_C:JP SAVE3
47F4 C3 65 47
47F7 ED 5B 10 30 3E 04 CD  144             LD DE,(linebf):LD A,4:CALL #FILE
47FE A3 1F
4800 CD 42 43 C3 6E 47     145             CALL CR_C:JP SAVE4
4806                       146 ;------------------------
4806                       147 SEARCH  : ストリングス サーチ
4806 CD E9 35 CD 32 36     148             CALL SPSKIP:CALL CHKNUM
480C 38 06 CD 12 36 22 5D  149             IF NC THEN CALL GETNUM:LD (lineno),HL
4813 30
4814 CD E9 35              150             CALL SPSKIP
4817 2A 22 30 CD C5 48 D8  151             LD HL,(seachbf):CALL CHRSET:RET C
481E CD 3B 34              152             CALL LINSET
4821 2A 5D 30 E5           153             LD HL,(lineno):PUSH HL
4825 CD 84 32 CD EC 34     154              CALL WINDPOP:CALL ENDCHK
482B E1 22 5D 30           155             POP HL:LD (lineno),HL
482F CD 97 33              156             CALL WTXTPRT
4832 CD 83 48 20 2C        157 SEARCH2 CALL SEARCHSB:JR NZ,SEARCH3
4837 AF 32 2E 30 CD E0 39  158             XOR A:LD (l_offs),A:CALL OFFSTR
483E CD B5 37 CD D1 37     159             CALL PCURDAT:CALL PLINDAT
4844 CD 3B 34 CD AA 33     160             CALL LINSET:CALL TXTPRT
484A CD 59 46 CD A0 37     161             CALL EDCUR:CALL INKEYCUR
4850 CD A9 36 FE 1B CA 78  162             CALL INKEYSUB:IF A=1BH JP COMLVLC
4857 41
4858 FE 20 CA 21 46        163             IF A=" " JP EDIT1
485D 21 7E 30 34           164             LD HL,pointer:INC (HL)
4861 18 CF                 165             JR SEARCH2
4863 AF 32 7E 30 32 2E 30  166 SEARCH3 XOR A:LD (pointer),A:LD (l_offs),A
486A 2A 5D 30 23 22 5D 30  167             LD HL,(lineno):INC HL:LD (lineno),HL
4871 ED 5B 5F 30 CD 76 34  168             LD DE,(lineadr):CALL NXTLIN
4878 ED 53 5F 30 1A        169             LD (lineadr),DE:LD A,(DE)
487D B7 20 B2              170             IF A<>0 JR SEARCH2
4880 C3 78 41              171             JP COMLVLC
4883                       172
4883                       173 SEARCHSB ;現在の行のpointer以後をサーチ
4883 2A 5F 30 3A 7E 30 4F  174             LD HL,(lineadr):LD A,(pointer):LD C,A
488A 85 6F 30 01 24        175             ADD A,L:LD L,A:IF C THEN INC H
488F EB                    176             EX DE,HL
4890 2A 22 30 CD 94 1F     177             LD HL,(seachbf):CALL #PEEK
4896 EB                    178             EX DE,HL
4897                       179 SEARCHSB1
4897 F5                    180             PUSH AF
4898 BE 20 0C              181              IF A<>(HL) JR SEARCHSB2
489B CD B3 48 20 07        182              CALL ?SAME:JR NZ,SEARCHSB2
48A0 79 32 7E 30           183              LD A,C:LD (pointer),A
48A4 F1 AF C9              184             POP AF:XOR A:RET
48A7                       185 SEARCHSB2
48A7 7E 23 0C              186              LD A,(HL):INC HL:INC C
48AA FE 20 38 03 F1 18 E6  187              IF A>=20H THEN POP AF:JR SEARCHSB1
48B1 D1 C9                 188             POP DE:RET
48B3                       189 ;
48B3                       190 ?SAME   ;文字列照合
48B3                       191             ; in  DE=string adr
48B3                       192             ; out flog Z=same
48B3                       193             ; des Acc,HL,DE
48B3 EB 2A 22 30           194             EX DE,HL:LD HL,(seachbf)
48B7 D5                    195             PUSH DE
48B8 CD 94 1F              196 ?SAME1  CALL #PEEK
48BB EB BE                 197              EX DE,HL:CP (HL)
48BD EB 13 23              198              EX DE,HL:INC DE:INC HL
48C0 28 F6                 199              JR Z,?SAME1
48C2 E1                    200             POP HL
48C3 B7 C9                 201             OR A:RET
48C5                       202 ;
48C5                       203 CHRSET  :文字列をバッファに書き込む
48C5                       204             ;in  DE=string adr
48C5                       205             ;    HL=buff adr(S-OS work)
48C5                       206             ;out DE=string end adr+1
48C5                       207             ;des Acc,B,HL
48C5 1A 13                 208             LD A,(DE):INC DE
48C7 FE 22 20 18           209             IF A<>22H JR CHRSET3
48CB 06 7F                 210             LD B,127
48CD 1A FE 0D 28 0B        211 CHRSET1 LD A,(DE):IF A=0DH JR CHRSET2
48D2 FE 22 28 07           212             IF A=22H JR CHRSET2
48D6 CD 9A 1F 13 23        213             CALL #POKE:INC DE:INC HL
48DB 10 F0                 214             DJNZ CHRSET1
48DD AF CD 9A 1F           215 CHRSET2 XOR A:CALL #POKE
48E1 B7 C9                 216             OR A:RET
48E3 CD 94 1F              217 CHRSET3 CALL #PEEK
48E6 B7 37 C8              218             OR A:SCF:RET Z
48E9 B7 C9                 219             OR A:RET
48EB                       220 ;------------------------
48EB                       221 CHENGE  : ストリングス置き換え
48EB 2A 22 30 AF           222             LD HL,(seachbf):XOR A
48EF 47 32 1C 4A           223             LD B,A:LD (chengmd),A
48F3 CD 9A 1F 10 FB        224 CHENGE1 CALL #POKE:DJNZ CHENGE1
48F8 1A FE 2F 20 04 32 1C  225             LD A,(DE):IF A="/" THEN LD (chengmd),A:INC DE
48FF 4A 13
4901 CD E9 35 CD 32 36     226             CALL SPSKIP:CALL CHKNUM
4907 38 06 CD 12 36 22 5D  227             IF NC THEN CALL GETNUM:LD (lineno),HL
490E 30
490F CD E9 35              228             CALL SPSKIP
4912 2A 22 30 CD C5 48 D8  229             LD HL,(seachbf):CALL CHRSET:RET C
4919 2A 24 30 CD C5 48 D8  230             LD HL,(chengbf):CALL CHRSET:RET C
4920 CD 3B 34              231             CALL LINSET
4923 3A 1C 4A B7 28 11     232             LD A,(chengmd):IF A=0 JR CHENGE2
4929 2A 5D 30 E5           233             LD HL,(lineno):PUSH HL
492D CD 84 32 CD EC 34     234              CALL WINDPOP:CALL ENDCHK
4933 E1 22 5D 30           235             POP HL:LD (lineno),HL
4937 CD 97 33              236             CALL WTXTPRT
493A CD 83 48 20 41        237 CHENGE2 CALL SEARCHSB:JR NZ,CHENGE5
493F 3A 1C 4A B7 28 27     238             LD A,(chengmd):IF A=0 JR CHENGE3
4945 AF 32 2E 30 CD E0 39  239             XOR A:LD (l_offs),A:CALL OFFSTR
494C CD B5 37 CD D1 37     240             CALL PCURDAT:CALL PLINDAT
4952 CD 3B 34 CD AA 33     241             CALL LINSET:CALL TXTPRT
4958 CD 59 46 CD A0 37     242             CALL EDCUR:CALL INKEYCUR
495E CD A9 36 FE 1B 28 38  243             CALL INKEYSUB:IF A=1BH JR CHENGE6
4965 FE 20 20 11           244             IF A<>" " JR CHENGE4
4969 CD C4 1F              245             CALL #BELL
496C CD 26 35              246 CHENGE3 CALL LINTRS
496F CD A3 49 30 06        247             CALL CHENGE7:JR NC,CHENGE4
4974 CD 80 36 C3 9D 49     248             CALL MEMOVR:JP CHENGE6
497A 21 7E 30 34           249 CHENGE4 LD HL,pointer:INC (HL)
497E 18 BA                 250             JR CHENGE2
4980 AF 32 7E 30 32 2E 30  251 CHENGE5 XOR A:LD (pointer),A:LD (l_offs),A
4987 2A 5D 30 23 22 5D 30  252             LD HL,(lineno):INC HL:LD (lineno),HL
498E ED 5B 5F 30 CD 76 34  253             LD DE,(lineadr):CALL NXTLIN
4995 ED 53 5F 30 1A        254             LD (lineadr),DE:LD A,(DE)
499A B7 20 9D              255             IF A<>0 JR CHENGE2
499D CD 87 33 C3 78 41     256 CHENGE6 CALL SCRNINIT:JP COMLVLC
49A3                       257
49A3 2A 22 30 CD 0D 4A EB  258 CHENGE7 LD HL,(seachbf):CALL BFCONT:EX DE,HL
49AA 2A 24 30 CD 0D 4A 44  259             LD HL,(chengbf):CALL BFCONT:LD BC,HL
49B1 4D
49B2 B7 ED 52 28 43        260             SUB HL,DE:JR Z,CHENGE10
49B7 38 1F                 261                      :JR C,CHENGE8
49B9 EB 2A 10 30           262             EX DE,HL:LD HL,(linebf)
49BD 01 FE 00 09           263             LD BC,0FEH:ADD HL,BC
49C1 E5 E5                 264             PUSH HL:PUSH HL
49C3 B7 ED 52 EB           265               SUB HL,DE:EX DE,HL
49C7 CD C4 39 44 4D        266               CALL PINADR:LD BC,HL
49CC E1                    267              POP HL
49CD B7 ED 42 44 4D        268              SUB HL,BC:LD BC,HL
49D2 EB                    269              EX DE,HL
49D3 D1 ED B8              270             POP DE:LDDR
49D6 18 22                 271             JR CHENGE10
49D8 E5                    272 CHENGE8 PUSH HL
49D9 CD C4 39              273              CALL PINADR
49DC EB 19                 274              EX DE,HL:ADD HL,DE
49DE EB 09                 275              EX DE,HL:ADD HL,BC
49E0 E5                    276              PUSH HL
49E1 2A 10 30 01 FE 00 09  277               LD HL,(linebf):LD BC,0FEH:ADD HL,BC
49E8 B7 ED 52 44 4D 03     278               SUB HL,DE:LD BC,HL:INC BC
49EE E1 EB ED B0           279              POP HL:EX DE,HL:LDIR
49F2 C1 EB                 280             POP BC:EX DE,HL
49F4 36 20 23              281 CHENGE9 LD (HL)," ":INC HL
49F7 0C 20 FA              282             IF INC(C)<>0 JR CHENGE9
49FA CD C4 39 EB           283 CHENGE10 CALL PINADR:EX DE,HL
49FE 2A 24 30              284             LD HL,(chengbf)
4A01 CD 94 1F B7 CA 4D 35  285 CHENGE11 CALL #PEEK:IF A=0 JP WRTLIN0
4A08 12 13 23              286             LD (DE),A:INC DE:INC HL
4A0B 18 F4                 287             JR CHENGE11
4A0D                       288
4A0D                       289 BFCONT  ;バッファ内の文字数
4A0D                       290             ; in  HL=buf adr
4A0D                       291             ; out C=string size
4A0D                       292             ; des Acc,HL
4A0D 0E 00 CD 94 1F        293             LD C,0:CALL #PEEK
4A12 B7 20 03 69 67 C9     294             IF A=0 THEN LD L,C:LD H,A:RET
4A18 23 0C 18 F3           295             INC HL:INC C:JR BFCONT+2
4A1C                       296
4A1C 00                    297 chengmd DB 0
4A1D                       298 ;------------------------
4A1D                       299 DELTXT  : テキスト消去
4A1D                       300
4A1D CD E9 35              301             CALL SPSKIP
4A20 CD 32 36 D8           302             CALL CHKNUM:RET C
4A24 CD 12 36              303             CALL GETNUM
4A27 E5                    304             PUSH HL
4A28 CD E9 35 CD 32 36     305              CALL SPSKIP:CALL CHKNUM
4A2E 30 02 E1 C9           306              IF C THEN POP HL:RET
4A32 CD 12 36 23           307              CALL GETNUM:INC HL
4A36 D1 EB                 308             POP DE:EX DE,HL
4A38 22 7A 30              309             LD (markbf),HL
4A3B CD 04 35 EB           310             CALL LINADR:EX DE,HL
4A3F CD 04 35              311             CALL LINADR
4A42 E5 B7 ED 52 E1        312             PUSH HL:SUB HL,DE:POP HL
4A47 C8 D8                 313             RET Z:RET C
4A49 7E                    314             LD A,(HL)
4A4A B7 20 03 12 18 10     315             IF A=0 THEN LD (DE),A:JR DELTXT1
4A50 E5                    316             PUSH HL
4A51 44 4D 2A 65 30        317              LD BC,HL:LD HL,(endadr)
4A56 B7 ED 42 44 4D        318              SUB HL,BC:LD BC,HL
4A5B E1 D8                 319             POP HL:RET C
4A5D 03 ED B0              320             INC BC:LDIR
4A60 2A 7A 30 22 5D 30     321 DELTXT1 LD HL,(markbf):LD (lineno),HL
4A66 CD 2C 33 CD 87 33     322             CALL TXTPUSH:CALL SCRNINIT
4A6C C3 78 41              323             JP COMLVLC
4A6F                       324 ;------------------------
4A6F                       325 SWPTXT  :テキスト交換
4A6F 1A 13 CD F0 35        326             LD A,(DE):INC DE:CALL UPPER
4A74 D6 41 DA C4 1F        327             SUB "A":JP C,#BELL
4A79 FE 04 D2 C4 1F        328             IF A>=4 JP #BELL
4A7E 32 3C 4B 4F           329             LD (SWPTXW1),A:LD C,A
4A82 CD E9 35 CD F0 35     330             CALL SPSKIP:CALL UPPER
4A88 D6 41 DA C4 1F        331             SUB "A":JP C,#BELL
4A8D FE 04 D2 C4 1F        332             IF A>=4 JP #BELL
4A92 B9 CA C4 1F           333             IF A=C JP #BELL
4A96 32 3D 4B              334             LD (SWPTXW1+1),A
4A99 CD CC 34 DA C4 1F     335             CALL TEXTDAT:JP C,#BELL
4A9F 22 40 4B ED 43 44 4B  336             LD (SWPTXW2+2),HL:LD (SWPTXW3+2),BC
4AA6 3A 3C 4B CD CC 34 DA  337             LD A,(SWPTXW1):CALL TEXTDAT:JP C,#BELL
4AAD C4 1F
4AAF 22 3E 4B ED 43 42 4B  338             LD (SWPTXW2),HL:LD (SWPTXW3),BC
4AB6 2A 44 4B AF ED 42     339             LD HL,(SWPTXW3+2):XOR A:SBC HL,BC
4ABC 38 01 3C              340             IF NC THEN INC A
4ABF 2A 3E 4B              341             LD HL,(SWPTXW2)
4AC2 B7 28 09 2A 40 4B ED  342             IF A<>0 THEN LD HL,(SWPTXW2+2):LD BC,(SWPTXW3+2):LD DE,B
C
4AC9 4B 44 4B 50 59
4ACE B7 20 06 ED 5B 44 4B  343             IF A=0  THEN LD DE,(SWPTXW3+2) ELSE LD DE,(SWPTXW3)
4AD5 18 04 ED 5B 42 4B
4ADB E5                    344             PUSH HL
4ADC AF ED B1              345              XOR A:CPIR
4ADF C1 C2 C4 1F           346             POP BC:JP NZ,#BELL
4AE3 B7 ED 42 EB           347             SUB HL,BC:EX DE,HL
4AE7 B7 ED 52 DA C4 1F     348             SUB HL,DE:JP C,#BELL
4AED 2A 42 4B 54 5D        349             LD HL,(SWPTXW3):LD DE,HL
4AF2 ED 4B 44 4B B7 ED 42  350             LD BC,(SWPTXW3+2):SUB HL,BC
4AF9 30 02 42 4B           351             IF C THEN LD BC,DE
4AFD                       352             ;
4AFD 2A 3E 4B ED 5B 40 4B  353             LD HL,(SWPTXW2):LD DE,(SWPTXW2+2)
4B04 7E 08                 354 SWPTXT1 LD A,(HL):EX AF,AF'
4B06 1A 77                 355             LD A,(DE):LD (HL),A
4B08 08 12                 356             EX AF,AF':LD (DE),A
4B0A 23 13                 357             INC HL:INC DE
4B0C 0B 78 B1 20 F3        358             IF DEC(BC)<>0 JR SWPTXT1
4B11                       359             ;
4B11 3A 3C 4B CD B0 44     360             LD A,(SWPTXW1):CALL FNADR
4B17 E5                    361             PUSH HL
4B18 3A 3D 4B CD B0 44     362              LD A,(SWPTXW1+1):CALL FNADR
4B1E D1 06 20              363             POP DE:LD B,20H
4B21 CD 94 1F 08           364 SWPTXT2 CALL #PEEK:EX AF,AF'
4B25 EB                    365             EX DE,HL
4B26 CD 94 1F 08 CD 9A 1F  366             CALL #PEEK:EX AF,AF':CALL #POKE
4B2D EB                    367             EX DE,HL
4B2E 08 CD 9A 1F           368             EX AF,AF':CALL #POKE
4B32 13 23 10 EB           369             INC DE:INC HL:DJNZ SWPTXT2
4B36 CD 52 33 C3 78 41     370             CALL SCRNINIT_c:JP COMLVLC
4B3C 00 00                 371 SWPTXW1 DB 0,0
4B3E 00 00 00 00           372 SWPTXW2 DW 0,0
4B42 00 00 00 00           373 SWPTXW3 DW 0,0
4B46                       374 ;------------------------
4B46                       375 PRWSW   :ウィンドウプライオリティ切換
4B46 CD FC 33              376             CALL PRWINIT
4B49 3A D9 30 B7 28 06     377             LD A,(wprwsw):IF A=0 JR PRWSW1
4B4F AF 11 E8 31 18 04     378             XOR A:LD DE,prwofm:JR PRWSW2
4B55 3C 11 E0 31           379 PRWSW1  INC A:LD DE,prwonm
4B59 32 D9 30 CD BA 42     380 PRWSW2  LD (wprwsw),A:CALL COM#MSX
4B5F CD C3 42 C3 42 43     381             CALL COM#PRT:JP CR_C
```

日本ソフトバンクの
書籍特約書店

下記の書店の一覧は、日本ソフトバンク書籍特約店として右にある商品の他、新刊もとりそろえております。ご希望の商品がある場合は、下記のお近くの書店にてお買い求め下さい。
(注)現品が売れて補充中の場合もございますので、ご注意下さい。

日本ソフトバンク出版事業部
〒102 東京都千代田区九段南2-3-26 ☎03(261)4095

全国特約書店一覧

地域	書店名	電話番号
〈北海道〉		
札幌市	紀伊國屋書店札幌店	011-231-2131
〃	旭屋書店札幌店	011-241-3007
〃	丸善札幌支店	011-241-7252
〃	リーブルなにわ	011-221-3800
〃	冨貴堂札幌パルコ店	011-214-2303
〃	ダイヤ書房本店	011-712-2541
〃	ダイヤ書房西店	011-665-6223
旭川市	旭川冨貴堂	0166-26-3481
〃	ブックス平和マルカツ店	0166-23-6211
苫小牧市	旭屋書店苫小牧店	0144-36-5185
〈東北〉		
青森市	成田本店	0177-23-2431
〃	岡田書店	0177-23-1381
弘前市	紀伊國屋書店弘前店	0172-36-4511
〃	ブックイン城東	0172-28-2882
八戸市	伊吉書院	0178-44-1917
盛岡市	東山堂ブックセンター	0196-53-6464
〃	さわや書店	0196-53-4411
〃	第一書店	0196-53-3355
仙台市	金港堂	022-225-6521
〃	金港堂ブックセンター	022-223-0979
〃	アイエ書店駅前店	022-264-0718
〃	丸善仙台支店	022-266-1127
〃	髙山書店	022-263-1511
〃	ブックスみやぎ	022-267-4422
秋田市	三浦書店	0188-33-8131
山形市	八文字屋	0236-22-2150
福島市	岩瀬書店コルニエツタヤ店	0245-21-2101
〃	博向堂	0245-21-1161
郡山市	東北書店	0249-32-0379
いわき市	ヤマニ書房本店	0246-23-3481
〃	鹿島ブックセンター	0246-28-2222
〈関東〉		
水戸市	川又書店駅前店	0292-31-0102
〃	ツルヤブックセンター	0292-25-2711
勝田市	武石書店	0292-73-1212
東海村	大野書店	0292-82-2098
鹿島郡	なみき書店	0299-96-1855
土浦市	共栄堂	0298-21-6134
筑波市	丸善筑波大学会館店	0298-51-6000
〃	友朋堂吾妻本店	0298-52-3665
宇都宮市	落合書店オリオン店	0286-34-3777
〃	落合書店東武ブックセンター	0286-34-8271
〃	新星堂宇都宮店	0286-33-2337
小山市	進駸堂駅ビル店	0285-25-1522
前橋市	煥乎堂	0272-23-1211
〃	リブロ前橋店	0272-34-1011
〃	戸田書店前橋店	0272-61-5063
高崎市	学陽書房	0273-23-4055
〃	サカ井書店	0273-62-1500
〃	新星堂高崎店	0273-27-3961
〃	戸田書店高崎店	0273-63-5110
太田市	ナカムラヤ	0276-22-2001
〈首都圏〉		
浦和市	須原屋本店	0488-22-5321
浦和市	須原屋コルソ店	0488-24-5321
大宮市	押田謙文堂	0486-41-3141
〃	ブックセンター押田	0486-47-3141
〃	三省堂ブックポート	0486-46-2600
蕨市	須原屋蕨店	0484-44-1211
川口市	岩渕書店川口店	0482-52-2190
川越市	黒田書店川越店	0492-25-3138
所沢市	芳林堂所沢店	0429-25-5355
〃	いけだ書店所沢店	0429-28-3271
上福岡市	黒田書店上福岡店	0492-66-0120
朝霞市	文教堂朝霞店	0484-76-0107
志木市	新星堂志木店	0484-74-0182
春日部市	文教堂春日部店	0487-52-7666
比企郡	錦電サービス	0492-96-2962
千葉市	多田屋セントラルプラザ店	0472-24-1333
〃	キディランド千葉店	0472-25-2011
習志野市	巌翠堂	0474-72-5011
船橋市	ときわ書房本店	0474-24-0750
〃	リブロ船橋店	0474-25-0111
〃	旭屋書店船橋店	0474-24-7331
〃	芳林堂津田沼店	0474-78-3737
〃	第二巌翠堂	0474-65-0926
柏市	西口アサノ	0471-44-2111
〃	新星堂柏店	0471-64-8551
松戸市	堀江良文堂	0473-65-5121
〃	辰正堂駅ビル店	0473-64-7997
横浜市	有隣堂トーヨー店	045-311-6265
〃	有隣堂東口ルミネ店	045-453-0811
〃	栄松堂相鉄ジョイナス店	045-321-6831
〃	そごうブックセンター	045-465-2111
〃	丸善ブックメイツポルタ店	045-453-6811
〃	有隣堂伊勢佐木店	045-261-1231
〃	有隣堂戸塚店	045-881-2661
〃	文華堂戸塚店	045-864-5151
〃	アーバン文華堂	045-821-5151
〃	文教堂青葉台南口店	045-983-5150
川崎市	有隣堂アゼリア店	044-245-1231
〃	有隣堂川崎BE店	044-200-6831
〃	文学堂本店	044-244-1251
〃	ブックセンター文教堂	044-811-5557
〃	文教堂溝の口店	044-811-8258
鎌倉市	島森書店大船店	0467-46-3841
〃	鎌倉書店	0467-46-2619
横須賀市	平坂書房WALK店	0468-25-5537
藤沢市	有隣堂藤沢店	0466-26-1411
〃	リブロ藤沢店	0466-27-0111
〃	文教堂六会店	0466-82-9610
茅ヶ崎市	川上書店ルミネ店	0467-87-3827
平塚市	サクラ書店駅ビル店	0463-23-2751
〃	文教堂四之宮店	0463-54-2880
小田原市	八小堂書店	0465-22-7111
〃	伊勢治書店	0465-22-1366
〃	文教堂小田原店	0465-36-3677
厚木市	有隣堂厚木店	0462-23-4111
大和市	文教堂中央林間店	0462-75-4165
相模原市	文教堂相模大野店	0427-49-0650
相模原市	文教堂橋本店	0427-74-5581
〃	文教堂星ヶ丘店	0427-58-6121
津久井郡	文教堂城山店	0427-82-9278
〈東京〉		
千代田区	三省堂書店神田本店	03-233-3312
〃	書泉グランデ	03-295-0011
〃	東京堂書店	03-291-5181
〃	旭屋書店水道橋店	03-294-3781
〃	丸善お茶の水店	03-295-5581
〃	巌翠堂	03-291-1362
〃	いずみ神田南口店	03-254-8521
〃	明正堂秋葉原店	03-257-0758
中央区	八重洲ブックセンター	03-281-1811
〃	日本橋丸善	03-272-7211
〃	旭屋書店銀座店	03-573-4936
港区	書原新橋店	03-591-8738
〃	雄峰堂ＮＳ店	03-503-6586
〃	虎ノ門書房本店	03-502-3461
〃	虎ノ門書房田町店	03-454-2571
品川区	芳林堂大井町店	03-474-4946
〃	明屋書店五反田店	03-492-3881
渋谷区	紀伊國屋書店渋谷店	03-463-3241
〃	旭屋書店渋谷店	03-476-3971
〃	三省堂書店渋谷店	03-407-4545
〃	大盛堂書店	03-463-0511
〃	紀伊國屋書店笹塚店	03-485-0131
新宿区	紀伊國屋書店本店	03-354-0131
〃	三省堂書店新宿西口店	03-343-4871
〃	福家書店センタービル店	03-345-1246
〃	福家書店野村ビル店	03-342-0298
〃	新星堂ＮＳビル店	03-344-2055
〃	西武新宿ブックセンター	03-208-0380
〃	芳林堂高田馬場店	03-208-0241
〃	未来堂	03-200-9185
豊島区	旭屋書店池袋店	03-986-0311
〃	芳林堂池袋店	03-984-1101
〃	リブロ池袋店	03-981-0111
〃	三省堂書店池袋店	03-987-0511
〃	新栄堂本店	03-984-2345
〃	新栄堂アルパ店	03-988-0181
台東区	明正堂中通り店	03-831-0191
墨田区	リブロ錦糸町店	03-846-0111
〃	ブックストア・談	03-635-1841
江東区	新栄堂亀戸駅ビル店	03-638-2345
江戸川区	文教堂西葛西店	03-689-3621
大田区	アクトブックスサンカマタ店	03-735-1551
〃	竜文堂大森駅ビル店	03-775-3851
中野区	明屋書店東京本社	03-387-8451
杉並区	ブックセンター荻窪	03-393-5571
〃	書原杉並店	03-313-4778
武蔵野市	紀伊國屋書店吉祥寺東急店	0422-21-5543
〃	弘栄堂吉祥寺店	0422-22-1031
〃	パルコブックセンター吉祥寺	0422-21-8122
調布市	真光書店	0424-87-2222
府中市	啓文堂	0423-66-3151
三鷹市	三省堂書店三鷹店	0422-48-4510

展示図書一覧

書名	価格
MS-DOSいたれりつくせり本	●1800円
プレイMS-DOS	●1900円
UNIX System V プログラマ・ガイド	●12000円
UNIX System V ユーザ・ガイド	●9800円
C言語の活用理解	●2000円
C言語の基礎知識	●2500円
C言語の応用50例	●2300円
Cプリプロセッサ・パワー	●2200円
Play the C 上巻	●1500円
Play the C 下巻	●1500円
8086アセンブリ言語	●2800円
8086マクロプログラミング	●2600円
ビギニングMUMPS	●2600円
マシン語マジックブックII	●2500円
マシン語プログラミングテクニック	●2000円
BASICによるプログラミングスタイルブック	●1800円
ソーティング・ノート	●1900円
BASICプログラムジェネレータ集	●2800円
98/88スモールビジネスプログラム集	●2500円
88デスクアクセサリ集	●2000円
IDOS活用ハンドブック	●2700円
DISK CHARGE追補版	●1800円
フロッピーディスクフル活用ガイド	●2300円
PC工作入門	●1800円
試験に出るX1	●2800円
X1テクニカルマスター	●2500円
X1システム研究室	●2500円
新松ガイド	●2000円
一太郎Ver.3ガイド	●2500円
新一太郎ガイド	●2300円
一太郎ガイド	●2000円
桐Ver.2ガイド	●2500円
花子応用ガイド	●2500円
Lotus 1-2-3ガイド	●2400円
RDBファラオガイド	●2900円
ビジュアルラーニングRDB	●2500円
アセンブラCASL入門	●2000円
ハードウェア徹底マスター	●2500円
FORTRAN徹底マスター	●2800円
特種情報処理試験 総整理と徹底対策	●2300円
情報処理の基礎知識	●1600円
ワープロ文書F・O・P	●1200円
新聞記事ハイテク切抜き法	●1200円
バイト&ワードの風にのって	●1800円
ワープロ考現学	●1200円
電子ゲームの「快楽」	●1200円
ムーグ・ノイマン・バッハ	●1300円
RPG幻想事典	●1500円
新明解ナム語事典	●5000円
保存版GS倶楽部	●1900円

市	書店	電話
三鷹市	東西書房	0422-46-0275
小金井市	文教堂小金井店	0423-86-0161
国分寺市	三成堂国分寺店	0423-25-3211
国立市	東西書店	0425-75-5061
小平市	文教堂小平店	0423-43-9229
東村山市	文教堂東村山店	0423-96-1115
立川市	オリオン書房ウイル店	0425-27-2311
八王子市	くまざわ書店本店	0426-25-1201
町田市	有隣堂町田店	0427-23-3018
〃	久美堂本店	0427-25-1330
〃	久美堂小田急店	0427-27-1111
〃	久美堂東急ハンズ店	0427-28-2772
〃	文教堂鶴川店	0427-35-4117
〃	文教堂小川店	0427-96-1781
多摩市	くまざわ書店桜ヶ丘店	0423-37-2531
福生市	文教堂福生店	0425-53-7708
〈信越・北陸〉		
長野市	平安堂長野店	0262-26-4545
〃	長谷川書店	0262-26-2122
上田市	平安堂上田店	0268-22-4545
松本市	ブックスロクサン	0263-35-5555
〃	改造社松本駅ビル店	0263-36-3777
飯田市	平安堂飯田店	0265-24-4545
岡谷市	笠原書店	0266-23-5070
諏訪郡	平安堂下諏訪店	0266-28-1111
新潟市	紀伊國屋書店新潟店	025-241-5281
〃	萬松堂	025-229-2221
〃	北光社	025-228-2321
長岡市	覚張書店	0258-32-1139
〃	ブックセンター長岡	0258-36-1360
〃	長岡技大長峰文化	0258-46-6437
富山市	瀬川書店	0764-24-4566
〃	清明堂	0764-24-4166
〃	BOOKSなかだ豊田店	0764-32-1353
〃	文苑堂本郷店	0764-22-0552
〃	文苑堂赤江店	0764-33-0321
高岡市	文苑堂	0766-21-0333
〃	文苑堂横田店	0766-21-0431
金沢市	うつのみや片町店	0762-21-6136
〃	書林香林坊本店	0762-20-5011
野々市町	王様の本本店	0762-46-5325
福井市	勝木書店	0776-24-0428
〃	品川書店新田塚店	0776-24-1112
〈東　海〉		
静岡市	静岡谷島屋呉服町本店	0542-54-1301
〃	江崎書店	0542-54-4481
〃	吉見書店	0542-52-0157
〃	戸田書店ＳＢＳ店	0542-81-5733
〃	戸田書店曲金店	0542-81-5899
沼津市	吉野屋	0559-23-5676
〃	マルサン書店宝塚店	0559-63-0350
富士市	戸田書店富士店	0545-51-5121
清水市	戸田書店本店	0543-65-2345
浜松市	浜松谷島屋連尺本店	0534-53-9121
名古屋市	三省堂書店名古屋店	052-562-0077
〃	星野書店近鉄ビル店	052-581-4796
名古屋市	丸善ブックメイツセントラルパーク	052-971-1231
〃	日進堂上前津店	052-263-0550
〃	三洋堂パソコンショップΣ	052-251-8334
〃	三洋堂いりなか本店	052-832-8202
〃	ちくさ正文館本店	052-741-1137
〃	白樺書房西店	052-774-7223
豊橋市	精文館	0532-54-2345
岡崎市	ブックス鎌倉	0564-54-1822
豊田市	三洋堂梅坪店	0565-35-2334
刈谷市	三洋堂刈谷店	0566-24-1134
春日井市	三洋堂勝川店	0568-32-7806
岐阜市	自由書房	0582-65-4301
大垣市	大洞堂ブックス258	0584-81-2553
〃	大洞堂岐大バイパス店	0584-74-7766
一宮市	三洋堂一宮店	0586-77-5734
可児市	三洋堂可児店	0574-63-2334
多治見市	三洋堂多治見店	0572-24-0340
津市	別所書店11ビル店	0592-24-1014
四日市市	文化センター白揚	0593-51-0711
鈴鹿市	シェトワ白揚スズカ	0593-82-5221
〈近　畿〉		
京都市	駸々堂京宝店	075-223-1003
〃	アバンティ・ブックセンター	075-682-5031
〃	オーム社書店河原町店	075-221-0280
〃	ジュンク堂京都店	075-252-0101
奈良市	駸々堂大丸店	0742-26-6241
大阪市	旭屋書店本店	06-313-1191
〃	紀伊國屋書店梅田店	06-372-5821
〃	オーム社書店大阪店	06-345-0641
〃	駸々堂京橋店	06-353-3209
〃	駸々堂心斎橋店	06-251-0881
〃	旭屋書店ナンバ店	06-644-2551
〃	ナンバブックセンター	06-644-5501
〃	旭屋書店アベノ店	06-631-6051
〃	ユーゴー書店	06-623-2341
〃	河村書店	06-951-2968
枚方市	水嶋書房京阪デパート店	0720-51-3432
高槻市	コーベブックス西武高槻店	0726-83-1766
東大阪市	ヒバリヤ書店本社	06-722-1121
神戸市	ジュンク堂センター街店	078-392-1001
〃	ジュンク堂サンパル店	078-252-0777
〃	海文堂書店	078-331-6501
〃	日東館書林	078-391-8701
姫路市	新興書房	0792-85-3344
〃	誠心堂書店	0792-81-2055
和歌山市	宮井平安堂	0734-31-1331
〃	帯伊書店	0734-22-0441
〈中　国〉		
岡山市	紀伊國屋書店岡山店	0862-32-3411
〃	丸善岡山支店	0862-31-2261
津山市	津山ブックセンター	08682-6-4047
広島市	紀伊國屋書店広島店	082-225-3232
〃	丸善広島支店	082-247-2251
〃	金正堂	082-248-3715
〃	積善館	082-248-3151
尾道市	啓文社尾道店	0848-37-5151
福山市	啓文社福山店	0849-22-3111
〃	ブックシティ啓文社	0849-25-0050
福山市	啓文社コア	0849-41-0909
山口市	五十部誠文堂	0839-24-6630
〃	文栄堂	0839-22-5611
宇部市	京屋書店	0836-31-2323
〃	末広書店	0836-31-0086
防府市	誠文堂国衙店	0835-25-1988
光市	三文字屋	0833-71-0251
鳥取市	富士書店	0857-23-7271
松江市	園山書店	0852-21-4167
〈四　国〉		
徳島市	小山助学館本店	0886-54-2135
〃	小山助学館東口店	0886-25-1380
〃	森住丸善	0886-23-3228
高松市	宮脇書店本店	0878-51-3733
丸亀市	宮脇書店丸亀店	0877-22-5533
松山市	紀伊國屋書店松山店	0899-32-0005
〃	明屋書店本店	0899-41-4141
〃	明屋書店大街道店	0899-41-4242
〃	丸三書店	0899-31-8501
新居浜市	明屋星原店	0897-44-4000
宇和島市	明屋宇和島店	0895-23-1118
高知市	金高堂	0888-22-0161
〈九州・沖縄〉		
福岡市	紀伊國屋書店福岡店	092-721-7755
〃	りーぶる天神	092-713-1001
〃	福岡金文堂	092-741-2106
〃	積文館新天町店	092-781-2991
〃	金文堂朝日ビル店	092-431-1094
北九州市	ナガリ書店	093-521-1044
〃	金栄堂	093-531-3685
〃	旭屋書店北九州店	093-631-6421
〃	井筒屋ブックセンター	093-641-0131
〃	カルパーク平野	093-661-7988
〃	白石書店本城店	093-601-2200
久留米市	エマックスたがみ	0942-33-1841
飯塚市	BOOKリード	0948-25-7266
大分市	パルコブックセンター大分店	0975-35-0643
〃	本町晃星堂	0975-33-0231
別府市	明林堂	0977-23-0936
宮崎市	田中書店中央店	0985-24-5111
〃	寿屋宮崎店	0985-27-4111
佐賀市	金華堂北バイパス店	0952-32-1965
〃	積文館デイトス店	0952-23-7155
長崎市	メトロ書店	0958-21-5453
〃	好文堂	0958-23-7171
佐世保市	金明堂	0956-22-4214
熊本市	紀伊國屋書店熊本店	0963-22-5531
〃	BOOKSまるぶん	0963-52-5665
〃	長崎書店	0963-53-0555
人吉市	明屋人吉店	0966-22-5486
鹿児島市	春苑堂ブックプラザ	0992-25-3200
〃	ブックスみすみ	0992-57-1011
那覇市	球陽堂書房ビル店	0988-63-3752
〃	文教図書	0988-62-1201

QUESTION and ANSWER

X1とMZ-700でS-OSの動作スピードがかなり（クロックの差以上の割合で）違うような気がするのですが，MZなどのS-OSもX1のハードにあわせるようなかたちで作られているのか，単にX1のハードが速いのか教えてください。　　東京都　井上　英治

ちょっと面白い質問なので取り上げてみました。確かにX1のS-OSは比較的速く，特にゲームにおいては圧倒的な速度を誇ります。しかし，S-OSはX1にあわせて作られているわけではありません。各機種ごとに最適な処理が行われているはずです。かといって，X1のハードが速いといってしまうのも語弊があります。「S-OSの実行に有利なハード構成をしている」程度の表現で抑えておきましょう。

X1が他機種と比べて特に速いのはキー入力です。多くの機種ではキースキャンをCPUが自ら行っているのに対して，X1ではサブCPUがキースキャンを行い，入力があれば割り込みをかけます。このような仕組みですので，リアルタイムキー入力はほかのマシンとは比べものにならないほど速くなっています。この差はいくつか発表されているシューティングゲームで特に目立つことになります。クロックの差を除けば，X1とMZ-700の速度の違いは，ほとんど，キー入力部分での差だと思われます。

また，MZ-700には当てはまりませんが，多くの機種ではVRAMをバンクメモリ上に置いていますので，文字の表示を行うためには，バンク切り換えをしなければなりません。X1ではVRAMはI/O空間に置かれていますので，このようなオーバーヘッドなしに文字の表示を行うことができます。今月発表されました「WINER」では画面表示を高速化するために，S-OSを介さないでテキストVRAMをアクセスするルーチンが各機種用に用意されますが，X1だけは専用ルーチンを使わなくても，そこそこの速度で動作するそうです。

勘違いされると困るので，つけ加えておきますと，一般にメモリに対するアクセスのほうがI/Oへのアクセスに比べて高速であり，その意味ではX1のようにI/O空間にVRAMを置いているマシンは絶対的な速度の点では不利です。たとえば，画面の全消去のような連続的なVRAMアクセスを行う際には，MZなどでも「バンクを切り換えっぱなし」にしますので，X1よりも数段速く画面を消すことができます。

結局，それぞれの方式に一長一短があるということで，一概にX1のハードが速いとはいえないということなのです。

X68000ユーザーです。AD PCMについてですが，一度に複数のデータを記憶させて，それを使い分けるにはどうすればいいのでしょうか。また，スペースハリアーのような3Dのスクロールはどうすればいいのでしょうか。PCMについては完璧に答えてください。あとは気が向いたらでいいです。BASICまたはアセンブラのリストつきで教えてください。　　石川県　阿部　貴秀

ひとつ目の質問は簡単です。ご希望どおりBASICのリストつきで解説しましょう。このプログラムはAD PCM1というファイルとAD PCM2という2つのファイルをそれぞれBUFF1，BUFF2で表される配列に読み込んでおいて，キー入力を待ち，スペースキーが押されたらADPCM1を，それ以外のキーならAD PCM2を鳴らします。気をつけなければいけないのは10行で宣言している配列の大きさを読み込むファイルの大きさに比べて十分大きく確保しておくことと，50行，80行で読み込むバイト数と140行，160行で実際に鳴らすバイト数を一致させる（後者が小さい分には可）ことぐらいです。難しいところはないと思いますので，あとはリストならびに今月のX68000BASIC入門を参考にしてください。

2番目の質問ですが，文面から察しますに，まだ阿部さんが踏み込んではならない領域ではないかと思えます。ご自分である程度試してみて，「ここまではできたのだが，あと一歩のところで行き詰まってしまった」というのであれば喜んでお答えするところですが，「試そうにもどこから手をつけていいのかわからない」のであれば，まだまだ質問するレベルには達していないということをご理解ください。

と文句をいいつつ，少し気が向いたので，簡単に解答しておきます。スペースハリアーの作り方です。まず，地面のスクロールですが，何色かのパレットコードで縞模様を描いておき，パレットを切り換えることで，それらしく見せることができます。たとえば，1～3のパレットコードで，

```
11111111111111111111111111111111
22222222222222222222222222222222
33333333333333333333333333333333
11111111111111111111111111111111
22222222222222222222222222222222
```

のような横縞を描いておいて，最初はパレットコード1と2が白，3が黒になるようにパレットを設定しておくとします。ここで，次の瞬間に2と3が白，1が黒になるようにパレットを切り換え，さらにその次の瞬間には3と1を白，2を黒に変更するといった処理を繰り返せば，縞模様が動いているように見えるでしょう。3Dにしたければ，一定の割合で，遠くは縞模様の間隔を狭く，近くは広くすればそれっぽく見えるようになります。

また，スペースハリアーのようにキャラクタの移動に伴って地平線を上下させたければ，X68000ならではのハードウェアスクロールで画面全体を上下に動かせばよいの

リスト1

```
 10 dim char buff1(1023),buff2(3839)
 20 int fp
 30 str s
 40 fp=fopen("adpcm1","r")
 50 fread(buff1,1024,fp)
 60 fclose(fp)
 70 fp=fopen("adpcm2","r")
 80 fread(buff2,3840,fp)
 90 fclose(fp)
100 while 1
110     s=inkey$(0)
120     if s="" then continue
130     if s=" " then {
140         a_play(buff1,4,3,1024)
150     } else {
160         a_play(buff2,4,3,3840)
170     }
180     while inkey$(0)<>"":endwhile
190 endwhile
```

です。パレットの切り換えと画面のハードウェアスクロールは非常に高速ですので，単にスクロールさせるだけのプログラムであれば，X-BASICで書いても速すぎるほどのものができます。

さて，たぶん，地面だけではなくキャラクタにも遠近をつけたいのでしょう。これも理屈は単純で，遠くのものほど小さく，近くのものほど大きくなるようにキャラクタを描くだけのことです。ひとつのキャラクタにつき，大きさを変えたいくつものパターンを用意して使い分けることになります。また，画面にキャラクタが複数存在するときには，重ね合わせが自然になるように遠くのものから順に描くようにします。

以上の処理はBASICでも書くことは可能です。もちろん，速度が非常に遅くなるのは目に見えていますし，仮にコンパイルしたとしてもスペースハリアーの域には到底及ばないでしょうが，アルゴリズムの確認の意味で，一度試してみるとよいでしょう。そこまでできれば，あとはマシン語で書き直し，ギンギンに最適化して，チラつきをなくす工夫，速度を一定にする工夫，データ量を減らす工夫などを加えれば，「あっという間」にスペースハリアーができあがります。（村田　敏幸）

よくX1用のゲームソフトなどでPSGで音声を出しているものがあります。これはどうしたらできるのですか。　神奈川県　岡本　浩

これはけっこう昔から行われている方法なのですが，最近のゲームソフトでよく使われているせいか，質問が多いのでお答えしましょう。いうまでもなくコンピュータはデジタルで動いているものですから，もともとアナログなものである音声をコンピュータから発生させるためには，アナログ(音声)→デジタル(コンピュータ)→アナログ(音源)という手順をふむことになります。

要するにコンピュータから音声を出すためにはAD（アナログ/デジタル）コンバータと，DA（デジタル/アナログ）コンバータが必要になるわけです。パソコンでもっとも簡単にこれを実現させるためには，ADコンバータとしてデータレコーダを，DAコンバータとしてPSGなどを用いるのがよいでしょう。すなわち，音楽の入ったテープを入れ，カセットからの入力が0（そのときの音量が規定値よりも低い）だったらPSGをオフ，入力が1だったらPSGをオンにしてやるのです。

これをX1用のプログラムにしてみたのがリスト2です。リスト2では入力されたテープの情報をそのままPSGへと流していますが，間にメモリをはさめば（テープ→メモリ→PSGとしてやる），自由に録音，再生ができることになります。

しかし，データレコーダというものは信頼性を上げるためハイパスフィルタを使って記録帯域を狭くしているものもあり，さらに0/1の1ビットでしかデータを扱わないので，音質は非常に悪いといえます。

音質を求めるためには，よりちゃんとしたADコンバータを用意し，それをうまくPSGで再生してやることでしょう。『Oh！FM』誌で採用されているHG-PLAY文では4ビットのADコンバータからサンプリングした音をPSGの0～15の音量に割り当ててサンプリングドラムなどを実現しています。また，ADコンバータ，DAコンバータともに専用のハードにしたものがPCM音源というわけです。（華門　真人）

リスト2

```
0000                 1 ;--------------------------
0000                 2 ; Voice by PSG
0000                 3 ;     for X1 CZ-8FB01(CB01)
0000                 4 ;--------------------------
0000                 5
D000                 6         ORG     0D000H
D000                 7
D000                 8 CMTCOM  EQU     00DECH
D000                 9 BRKCHK  EQU     0004AH
D000                10
D000                11 PORT2B  EQU     01A01H
D000                12 PSGADR  EQU     01C00H
D000                13
D000 AF             14         XOR     A           ;Init Ch.a
D001 67             15         LD      H,A
D002 CD 35 D0       16         CALL    PSG
D005 3C             17         INC     A
D006 CD 35 D0       18         CALL    PSG
D009 3E 07          19         LD      A,7
D00B 26 3E          20         LD      H,03EH      ;Ch.a ON
D00D CD 35 D0       21         CALL    PSG
D010                22
D010 01 01 1A       23         LD      BC,PORT2B
D013 3E 02          24         LD      A,002H      ;CMT play
D015 CD EC 0D       25         CALL    CMTCOM
D018                26
D018                27 CMTLOP
D018 CD 4A 00       28         CALL    BRKCHK
D01B 28 13          29         JR      Z,JOBEND    ;Break-> End
D01D ED 78          30         IN      A,(C)       ;CMT data
D01F E6 02          31         AND     002H
D021 26 08          32         LD      H,008H      ;Volume 8
D023 C2 28 D0       33         JP      NZ,LOOP1
D026 26 00          34         LD      H,000H      ;Volume 0
D028                35 LOOP1
D028 3E 08          36         LD      A,008H
D02A CD 35 D0       37         CALL    PSG
D02D C3 18 D0       38         JP      CMTLOP
D030                39
D030                40 JOBEND
D030 3E 01          41         LD      A,001H      ;CMT stop
D032 C3 EC 0D       42         JP      CMTCOM
D035                43
D035                44 PSG
D035 C5             45         PUSH    BC
D036 01 00 1C       46         LD      BC,PSGADR
D039 ED 79          47         OUT     (C),A       ;set reg.
D03B 05             48         DEC     B
D03C ED 61          49         OUT     (C),H       ;set data
D03E C1             50         POP     BC
D03F C9             51         RET
```

質問にお答えします

日ごろ疑問に思っていること，どんなことでも結構です。どんどんお便りください。難問，奇問，編集室が総力を上げてお答えいたします。ただし，お寄せいただいているものの中には，マニュアルを読めばすぐに回答が得られるようなものも多々あります。最低限，マニュアルは熟読しておきましょう。質問はなるべく具体的に機種名，システム構成，必要なら図も入れてこと細かに書いてください。また，返信用切手同封の質問をよく受けますが，原則として，質問には本誌上でお答えすることになっていますのでご了承ください。なお，質問の内容について，直接問い合わせることもありますので，電話番号も明記してくださいね。

宛先：〒102 東京都千代田区
九段南2-3-26井関ビル
㈱日本ソフトバンク出版部
「Oh！X質問箱」係

FILES Oh!X

このインデックスは，タイトル，注記——筆者名，誌名，月号，ページで構成されています。マイコンショウ，ビジネスショウで発表された新製品のニュースがにぎやかですね。

参考文献
I/O　工学社
ASCII　アスキー
The BASIC　技術評論社
テクノポリス　徳間書店
パソコンワールド　コンピュータワールド・ジャパン
POPCOM　小学館
マイコン　電波新聞社
マイコン BASIC Magazine　電波新聞社
LOGIN　アスキー

一般

▶マイコンショウ'88　第66回ビジネスショウ
平和島で開催されたマイコンショウと晴海のビジネスショウの主な見せ場を紹介。AX386やOS-9/X68000も写真入りで紹介されている。——編集部，I/O，7月号，170-173pp.

▶NEW MACHINE
シャープより発表された80386搭載マシン，AX386の仕様や同梱ソフトについての解説。——編集部，I/O，7月号，174p.

▶New Products
シャープから発売になったノートワープロ WV-500やビジネスファックス FO-420など OA 機器 5 種を紹介する。——編集部，I/O，7月号，297p.

▶ASCII EXPRESS シャープと三菱電機が並列処理が可能なデータ駆動型マイクロプロセッサを共同開発
シャープと三菱電機が発表したデータ駆動型マイクロプロセッサについて。——編集部，ASCII，7月号，145p.

▶ASCII EXPRESS シャープ，電子システム手帳用の IC カードを発売
シャープの電子システム手帳PA-7000用ICカード「シティガイド東京編カード」(PA-IC8)について。——編集部，ASCII，7月号，148p.

▶ASCII EXPRESS シャープが Bware シリーズに A4サイズの多機能日本語ワープロを投入
シャープから新しく発売される日本語ワープロWV-500の機能，価格についての紹介。——編集部，ASCII，7月号，151p.

▶ASCII EXPRESS 第66回ビジネスショウ
5月18日から4日間にわたって国際見本市でビジネスショウが開催された。そこで出品された新製品や参考出品についての紹介。——編集部，ASCII，7月号，154p.

▶最新機種レポート'88　32bit やラップトップも登場した AX
シャープの新製品 AX386など各社から発表された AX マシンが紹介されている。——編集部，ASCII，7月号，169-172pp.

▶キミにも同人誌ソフトが作れるんだよ
初心者からマシン語がわかる人までだれにでもソフトが作れる，という短期集中連載。今回はゲームデザインについて。——オニオン，テクノポリス，7月号，85-89pp.

▶RANDOMFILE
X68000用ワープロ EW の簡単な紹介。——編集部，POPCOM，7月号，149p.

▶RANDOMFILE
CPU に80286を採用した MZ-6500モデル50の機能を簡単に紹介。——編集部，POPCOM，7月号，149p.

▶パソコン入門講座
BASIC からファイルをランダムアクセスする方法を紹介。先月号のプログラムについても解説。——編集部，POPCOM，7月号，216-218pp.

▶シャープ AX386レポート
AX386のハードからソフトまで仕様の解説。——高橋雄一，マイコン，7月号，170-176pp.

▶電脳空間 RPG
ネットワークを使ったみんなで遊ぶ RPG。モニター募集も開始。——編集部，LOGIN，7月号，256-259pp.

MZ-80K/C/1200/700/1500

MZ-80K/C/1200/700/1500

▶CHANGE THE BOX
バラバラになった BOX を，元に戻すというパズルゲーム。——基克，マイコン BASIC Magazine，7月号，140-141pp.

MZ-700/1500

▶オリエンテーリング
ドラクエIIIの回転床をヒントにして作ったというゲーム。S-BASIC用。——まっぴー，マイコンBASIC Magazine，7月号，142-143pp.

▶THE LEGEND OF DRAGON
ドラクエタイプのRPG。敵キャラがよくできている。HuBASIC用。——カリット，マイコンBASIC Magazine，7月号，144-146pp.

MZ-1500

▶SUPER GOLF
全 9 ホールのゴルフコースで39人の相手とトーナメント大会。——山本進一，マイコン BASIC Magazine，7月号，147-149pp.

MZ-80B/2000/2500/2800

MZ-80B/2000/2500

▶Mole Game
昔懐かし，モグラたたきゲームです。——竹内守，マイコン BASIC Magazine，7月号，150-151pp.

▶UFO ATTACK
空を飛んでいるUFOを，レーザーで撃ち落とすゲーム。

この本の主題はひとつです。「人間は機械であり，行為のためにプログラミングされたロボットである」ということ。著者は人間という生物を肉体だけでなく，行為，思考，情緒や創造力でさえ，科学的に説明可能なひとつのシステムだといっています。ですから本書の内容は多岐にわたり，生物学，心理学，哲学など豊富な知識が読むものを圧倒します。確かに人間がそういう科学的に説明可能な一種のサイバネティックシステムだとしたら，それは機械であり，ロボットとなんら変わることはないでしょう。いまひとつ論理的でなかったりしますが，そのまま鵜呑みにしてしまうと，かなり衝撃的な内容です。

しかし，それはあくまでも理屈上の話であり，仮に心までが機械として説明できたとしても，それは遠い未来の話。そううまくいくなら人工知能研究者がこんなに困っているわけはありません。仮に心や行為の因果関係までわかり，チューリングテストに合格するような知能ができた日にはいまよりずっと住みにくい世の中になっていることを，私は断言しましょう。（K）

人間はロボットか
Geoff Simons 著　岡田弓子訳　オーム社刊
A5判変型　259ページ　2,600円　☎03(233)0641

敵の攻撃方法にいくつかバリエーションがある。――並木パソコンクラブ，マイコンBASIC Magazine，7月号，152-153pp.

MZ-2500

▶フリプシー

MZ-1500，X1版からの移植版パズルゲーム。――山岸秀匡，I/O，7月号，189-194pp.

▶ワタの鳴く夜は恐ろしい

動物園の珍獣ワタを，夜鳴かせないためにはどうするか，というパズルゲーム。全部で10面。――謎のパズル大好きおじさん，マイコンBASIC Magazine，7月号，154-156pp.

X1/X1turbo/Z

X1シリーズ

▶誌上RPG サンダーロード

テキストRPG第4章，ドラゴンの洞窟。――グループ・クラムボン，POPCOM，7月号，232-242pp.

▶誌上公開質問状　シャープ(X1)

X1turbo ZIIで2HD→2D，2D→2HDにプログラムを転送する方法など。――多田太郎，マイコンBASIC Magazine，7月号，72-73pp.

▶ Lip Lop Lap

敵をブラックホールに誘導して殺すゲーム。全10面。――Random田村　Live in福岡，マイコンBASIC Magazine，7月号，193-195pp.

▶おちゃはやっぱり！　北岡園！

画面上の5つの“あとらんだ”を集めるパズルゲーム。細かいルールが，パズルの面白味を出している。――数学倶楽部チームはりくんTM，マイコンBASIC Magazine，7月号，196-198pp.

X1turboシリーズ

▶ MS-DOS↔CP/Mファイルコンバータ

X1turboに2DD/2HD共用ドライブを接続してMS-DOS，CP/M間のデータの交換を実現するプログラム。CP/M80用Turbo Pascalが必要。――江上邦博，I/O，7月号，233-237pp.

▶モノクロ・ハードコピープログラム

エプソン製のプリンタVP-135Kの24ピン3倍密度モードを使ってモノクロ写真風にディスプレイ画面をハードコピーするプログラム。――古賀信一，I/O，7月号，259-263pp.

▶なんでもQ&A X1/X1turbo/X68000シリーズ編

turboCP/Mでのプリンタ設定の変更の仕方。――シャープ，マイコン，7月号，392-393pp.

X68000

▶涙なしのC言語講座

X68000用のC言語XCとPC-9801用の最新C言語Turbo Cのint型の表現やレジスタ変数などの比較。――吉沢正敏，I/O，7月号，139-141pp.

▶最新ソフトウェア情報

THE福袋V2.0を紹介している。――編集部，I/O，7月号，296p.

▶ X68000 WORKSHOP Information Shop

ビジネスショウでデモンストレーションの行われたOS-9/X68000について簡単に紹介する。――古谷野和彦，ASCII，7月号，261p.

▶ X68K Programmer's Shop

X68000用に登場したOS-9/X68000の概念，モジュール構造，カーネル等について。――古谷野和彦，ASCII，7月号，262-265pp.

▶ X68K Application Shop

正規表現でテキストファイル中の文字列検索を行うプログラムXFINDを掲載。――宮本親一郎，ASCII，7月号，266-268pp.

▶縮小印刷ソフト

プリンタに縮小漢字を印刷するプログラム。A4サイズの用紙でANK文字が横180×縦100文字以上入る。C compiler PRO-68Kが必要。――獨澄旻，The BASIC，7月号，78-82pp.

▶ X68000 2大新作ゲーム情報　ドラゴンスピリット

新作ゲーム「ドラゴンスピリット」の開発途中バージョンのエリア1，2画面写真の紹介。――編集部，テクノポリス，7月号，66-67pp.

▶ X68000 2大新作ゲーム情報　R-TYPE

アイレムが現在X68000用に移植中のアクションゲーム「R-TYPE」の開発中間報告。連続写真でステージ1のマップやステージ3の巨大戦艦の写真があり，「アーケード以上」といわれるグラフィックの美しさがわかる。――編集部，テクノポリス，7月号，68-70pp.

▶ THE福袋V2.0を開けてみる

福袋V2.0に含まれるアセンブラ，リンカなどの新バージョンの旧バージョンからの変更点を解説。――高橋雄一，マイコン，7月号，187-192pp.

▶ X68000マシン語入門

連載第10章。論理演算命令1つひとつについてサンプルフログラムつきで解説。またIOCSコールについてもBIOS番号表を載せている。――高橋雄一，マイコン，7月号，193-202pp.

▶ X68000ゲーム情報

X68000の新作ゲーム「ドラゴンスピリット」の開発途中バージョンの画面写真。――編集部，マイコン，7月号，217p.

▶ Z'sSTAFF PRO-68K

Z'sSTAFF使いこなし講座第3回。トーンやグラデーションについての解説。――紀要介，マイコン，7月号，263-266pp.

▶なんでもQ&A X1/X1turbo/X68000シリーズ編

X68000で使用可能である純正以外のプリンタについて。――シャープ，マイコン，7月号，392p.

▶なんでもQ&A X1/X1turbo/X68000シリーズ編

X68000でファンクションキーの内容を変える方法について。――シャープ，マイコン，7月号，393p.

▶誌上公開質問状　シャープ(X1)

X-BASICでプリンタにコントロールコードを送る場合などについて。――多田太郎，マイコンBASIC Magazine，7月号，72-73pp.

▶ X68K グランプリF1レース

マウスを使って車を操作するレーシングゲーム。――樋口裕司，マイコンBASIC Magazine，7月号，199-201pp.

▶ THE GAME MUSIC PROGRAM グラディウスII

ゲームミュージックプログラム。――Yu-You，マイコンBASIC Magazine，7月号，208-210pp.

▶ X68000新聞

今月のテーマは「眼にしみるグラフィック」。また，リターン・オブ・イシターなど新作ソフトの開発着手を紹介。――編集部，LOGIN，7月号，226-231pp.

ポケコン

PC-1360

▶漢字ミニデータベースプログラム

PC-1360の漢字BASICで作られたデータベースのプログラム。プログラムの説明もある。――塚田洋一，マイコン，7月号，360-364pp.

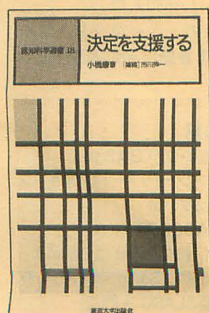
PC-1445，PC-E200

▶ CASL太鼓判

CASL入門講座の最終回。PUSH，POP，CALL，RET命令についての解説。――塚田洋一，マイコン，7月号，365-369pp.

PC-1500

▶ DRAGON1500

先月のアフターバーナーもどきに続いて，今月はドラゴンバスターもどきの登場。――桐山直己，マイコンBASIC Magazine，7月号，205-206pp.

決定を支援する

決定というのは至極一般的な日常行為である。本書で問われているのは，決定すること自体というより「よりよい意思決定を行う」ことであり，そのためには人間は習慣的な行為の一歩外に立つ必要があること，場合によっては問題の認知そのものを変えねばならないことを示している。コンピュータシステムは，こうした人間の意思決定を支援するもの，つまりデシジョン・エイドとしてその役割が考察されている。

小橋康章著　市川伸一補稿　東京大学出版会

A5判　222ページ　1,800円　☎03(811)8814

ニューラルネットワーク情報処理

「コネクショニズム入門，あるいは柔らかな記号に向けて」という副題のある本書は，高度並列分散情報処理を紹介する専門的なものである。

人間の脳における情報処理システムの原理が注目され始めて以来，情報理論の研究は並列処理の新しい可能性を探ってきた。ここでは，並列分散処理の基礎的な解説から，そのメカニズムと認知科学や人工知能分野への応用，さらに情報処理における「記号」とは何か，という問題に触れている。

麻生英樹著　産業図書刊

A5判　198ページ　2,200円　☎03(261)7821

バックナンバー案内

BACK ISSUES

ここには1987年８月号から1988年７月号までをご紹介しました。現在，1987年2, 3, 4, 5, 7, 8, 9, 10, 11, 12, 1988年1, 2, 3, 4, 5, 6, 7 までの在庫がございます。バックナンバーおよび定期購読のお申し込み方法については，本文172ページを参照してください。

1987

8月号

特集　迷宮の日本語処理環境
MZ-2500用ワープロプログラムSuperものかきくん
書式ユーティリティCOLN/らくらくSYMBOL他
試験に出るX1　最終回　通信プログラムである
X68000BASIC入門　第１回　めぐりあいX-BASIC
●X1/turbo用パズルゲーム　STAR PANIC
●Z'sSTAFF PRO 68Kの世界
X68000あなたの知らない世界　SOUND PRO-68K他
全機種共通システム　FM-7/77版S-OS"SWORD"他

9月号

特集1 MZ-700に不可能はない
MZ-700ゲームテクニック集/SPACE BLUSTER SG
特集2 ミュージックデータと遊ぶFM音源の世界
MZ-2500MMLの拡張/X1/turbo用MMLコンバータ
X68000あなたの知らない世界　マシン語入力ツール
BASICリレー連載　ディレクトリまるごとコピー
●X1turboZ，X68000用ハードコピープログラム
全機種共通システム　PC-80/88版S-OS"SWORD"
リロケータブル逆アセンブラInside-R

10月号

特集　Game Designを考える
遊びを設計するために/ピコピコゲームが原点 他
●投稿ゲーム４選
●ミュージックプログラム　ベートーベン月光
THE SOFTOUCH SPECIAL　イース/ウルティマⅣ
X68000あなたの知らない世界　BASIC to Cコンバータ
X68000BASIC入門　追撃ランダムファイル
全機種共通システム　FuzzyBASICコンパイラ拡張版
X1turbo版S-OS"SWORD"/tiny CORE WARS

11月号

特集1 全機種共通システムS-OS再考
超入門S-OS/ファイルアロケータ&ローダ
FuzzyBASICコンパイラ版BACK GAMMON
特集2 MZ-2500スペシャル 逆襲のアルゴ機能
アルゴブロック崩し/アルゴリズムを作ろう
●MZ-2500カードゲーム　KING'S COURT
THE SOFTOUCH　X68000用Kamikaze/MZ-2861用
upシリーズ/トリフォニー/リバイバー他
X68000あなたの知らない世界　CP/M-68K/TITLE. SYS

Oh!X　12月号

特集　正真正銘のOh! CZ SPECIAL
新製品速報X1turboZⅡ/X1twin/X68000
X1/turboシステム&プログラミング
NEW Z-BASIC/C compiler PRO-68K
人類タコ科図鑑　第１回　Jap meets Yankee
実用(?)オブジェクト指向のゲームプログラミング第1回
●X1/turbo用カードゲームSPEED
●X68000ファイルコンバータ　MACS/HELPS
全機種共通システム　PASOPIA7版S-OS"SWORD"他

1988

1月号

特集　MZ&X拡張ボードの活用
すべての道はI/Oに通じる/MZでX1用ボードを使う
1987年度GAME OF THE YEARノミネート発表
●MZ-2500用　ALGO SPACE BLUSTER SG
●LIVE in '88　ドラゴンスピリット/悲しきチェイサー
BASICリレー連載　半熟FORTRANはいかが
X68000BASIC入門　グラフィック炎上
マシン語体操1・2・3　データ構造を考えよう
全機種共通システム　Fuzzy BASICコンパイラ 奥村版

2月号

特集　グラフィック画像の冒険
X1/turboCGアニメ/トリフォニーで立体モデル
X68000グラフィックデータ/QUICK MZ PAINT他
X68000あなたの知らない世界　辞書構造/WORD POWER
マシン語体操1・2・3　Lispインタプリタ(1)
●NEW Z-BASIC詳報　その名はZ-BASIC
●LIVE in '88　グラディウス２
●SHORT ACCESS　THRILLING/POMカードポーカー
全機種共通システム　シューティングゲームELFES

3月号

特集　コンピュータサウンド"楽"入門
X1/turbo MIDIインタフェイスの製作
MZ-2500 Super Keyboard/VIPサウンドデータ公開
Oh!X LIVE SPECIAL 組曲「Ys」/Raspberry Dream他
THE SOFTOUCH　Might and Magic/Hyper UD
オブジェクト指向のゲームプログラミング
X68000BASIC入門　奇襲アニメ作戦
X68000あなたの知らない世界　未公開IOCSの解析
全機種共通システム　構造型コンパイラ言語SLANG

4月号

特集　不思議の国のゲーム学
決定！　1987年度GAME OF THE YEAR
ピコピコゲーム春場所/GAME REVIEW 10本他
新製品　X68000ACE-HD/カラースキャナCZ-8NS1
X68000あなたの知らない世界　microEMACSの移植
●MZ-700　SPACE BLUSTER FX
●LIVE in '88　Moonlight Serenade/Long Night他
全機種共通システム　デバッギングツールTRADE
シミュレーションウォーゲームWALRUS

5月号

特集　BASIC入門「再検証」
BASICの歴史と意義/栄光のHuBASIC
黄金のBASIC入門プログラム/プログラミング用語集
ミュージックプログラマへの道/レイトレーシング
特別企画　言わせてくれなくちゃだワ
●新製品　X68000ACE/ACE-HD
●LIVE in '88 GET WILD/BOOM BOOM/SDI
●SHORT ACCESS 3Dボクシング/マシン語データ文生成
全機種共通システム　シューティングゲームELFES

6月号　創刊6周年記念

特集　システム環境を考える
８ビットパソコンの開発環境/Human68kのシステム環境/システムを読むためのアセンブラ入門
特別企画　究極の８ビットパソコン　8RON計画
THE SOFTOUCH　X68000用日本語ワープロEW他
●付録「あぶない福袋」
マシン語体操1・2・3　番外編　Lisp80入門
X68000BASIC入門　捨て身のミュージック
全機種共通システム　構造化言語SLANG入門 他

7月号

特集　実践C言語からの誘惑
入門C言語/実録Cプログラミング/XBAS to C
THE SOFTOUCH　ソーサリアン/ゼリアード/アルギースの翼/SUPER大戦略/3大麻雀ソフト 他
●Oh!X LIVE in '88/SHORT ACCESS
新連載　C調言語講座PRO-68K　まずはprintfより始めよ
あなたの知らない世界　OS-9/X68000／Sampling PRO-68K
全機種共通システム　構造化言語SLANG 入門(2)
マルチウィンドウドライバMW-1

PENGUIN INFORMATION CORNER

ペ・ン・ギ・ン・情・報・コ・ー・ナ・ー

NEW PRODUCTS

カラービデオプリンタ
GZ-P21
シャープ

ビデオやテレビ，パソコンなどの映像情報をフルカラーで記録するビデオプリンタGZ-P21(198,000円)が，6月10日よりシャープから発売された。

プリントサイズは最大100×80mmで，縮小モード(79×60mm)と2画面モード（79×50mm×2画面）のプリントができる。また，デジタルマルチ機能により，1枚の印画紙に4画面/25画面のプリントをすることも可能。用途に応じてランニングコストの安い白黒シートも選択できる。

簡単なスイッチ切り換えひとつで，左右反転(ミラー)プリントモードにできる機能も持っている。

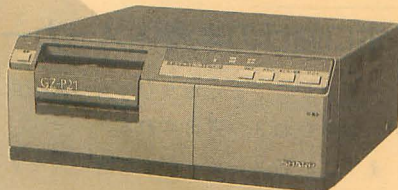
イエロー，シアン，マゼンタの3色面順次印画で，各色64階調，プリント画素数は縦478×横600ドット。

動画入力時は，内蔵の高画質デジタルフレーム/フィールドメモリにより，瞬時に静止画としてメモリし，プリントする。

アナログRGB(21ピン)，Sビデオ入力端子つき。

印画用シートインクセットは100枚で9,000円，ハガキ用20枚で3,500円，白黒シートは100枚で1,500円。

〈問い合わせ先〉
シャープ㈱　☎06(621)1221,03(260)1161

GZ-P21

カナ文字でデータ管理
電子カナメモPA-175/375
シャープ

シャープは，電子メモシリーズの新製品として，PA-175/375(各5,000円）を7月7日より発売した。

これらの電子メモは，カナ文字で最大160人分の名前と電話番号を記憶させられ，50音順に検索できる。データ1件につき最大60桁までの番号を入れられる。

ほかにも日付・時刻・用件などを記憶するスケジュール機能，データを秘密にするシークレット機能，記憶させたデータを呼び出して使うこともできる10桁計算機能などを備えている。

表示は12桁×2行でリチウム電池1個を使用。

PA-175は厚さ2.3mmという薄型で連続表示1500時間，PA-375は手帳タイプで連続表示2000時間。

〈問い合わせ先〉
シャープ㈱　☎06(621)1221,03(260)1161

PA-175/375

ワンタッチで自動ダイヤル
電子ダイヤラーPA-600
シャープ

シャープは，6月21日から電子ダイヤラーPA-600(12,800円)を発売した。

PA-600は，送話口に直接取りつけて信号を送るもので，市外局番を自動的にとばしてダイヤルするなど，ワンタッチで電話がかけられる。

また，新電電各社の回線を使い外出先などからクレジットコールをかけることもでき，この場合は自分のクレジットコール番号や相手先番号を記憶させて自動的にダイヤルできる。

PA-600

10桁1メモリの電卓機能や，最大約560人分の電話番号を記憶させておく電話帳機能もある。

表示部はカタカナ12文字×2行。サイズは幅66×奥行き139×厚さ11.4mm，重量93g。

〈問い合わせ先〉
シャープ㈱　☎06(621)1221,03(260)1161

アナログカラーディスプレイ
CU-2100/14ED
シャープ

シャープは，21インチのアナログカラーディスプレイCU-21CDをこの8月に発売する。価格は未定。入力信号周波数15/24/31kHz自動切り換え，3モードマルチスキャン方式採用。解像度は640ドット×200/400/512ラインでワイドスイッチ装備。

また，14型のアナログカラーディスプレイCU-14ED(79,800円）も発売中で，こちらは15/24kHz自動切り換え，2モードマルチスキャン方式採用，ワイドスイッチつき。

これのディスプレイは発売予定のシステ

CU-21CD

ムチューナーAN-8TU(35,800円) を接続するとテレビ受像機にもなる。対応機種はX1turboZ,X68000 (CU-14EDでは不可), PC-8801/9801シリーズなど。
〈問い合わせ先〉
シャープ(株) ☎06(621)1221, 03(260)1161

ノートワープロ新製品
ワードバンクノート2
セイコーエプソン

A4サイズのワードバンクノートがバージョンアップして登場。新製品ワードバンクノート2(74,800円)は6月14日にセイコーエプソンから発売された。重量1.2kg。

ワードバンクノート2

価格は5,000円アップして通信機能を標準装備し，プリンタの対応機種を増やした以外は最初のバージョンと同じもの。

通信機能としては，300から9600bpsに対応し，シフトJIS漢字コード採用，IDやパスワードを12カ所まで登録してオートログインできる。スクロールバッファは全角文字にして約6,000文字分。

辞書は複合語や固有名詞を含め約13万語，JIS第2水準漢字を装備，一括変換は最大200文字まで。

計算機能，住所録機能，スケジュール管理機能などを持ち，またワードラップやジャスティフィケーション機能があるので欧文ワープロとしても使用できる。

内部メモリは約18KバイトでA4サイズの原稿を約9ページストアできる。標準装備の外部記憶装置にはICカード(32Kバイト6,000円，8Kバイト3,000円)でデータを保存する。

STN液晶ディスプレイでガイドラインのほかに40文字×5行が表示できる。

プリンタケーブルは別売(5,000円)。
〈問い合わせ先〉
セイコーエプソン㈱ ☎0266(52)3131

プリンタ2種装備
キヤノンワードボーイPW-90
キヤノン

ワードボーイPW-90
本体プリンタとハンディプリンタの2つを標準装備した日本語ワープロ・キヤノンワードボーイPW-90(54,800円)がキヤノンより6月10日から発売された。

ハンディプリンタは，コピー感覚で印刷

Again Watch

恐慌が続く

先月号で，4MビットダイナミックRAM(DRAM)の話をしたが，ご承知のとおり，いま全世界的にメモリが極度に足りない。作っても作っても不足という状態で，メモリの大恐慌になっている。

これではとても4M時代の夢を見ている余裕はない。

品物別に見ると256KビットのDRAM，1MビットDRAM，256KビットのスタティックRAM (SRAM) などで，RAMが全般的に不足している。

早い話がニーズが高い先端商品が軒並み不足しているということである。この半導体不足，いろいろな現象を巻き起こし始めた。並べてみよう。

＊　＊

PC-9801が買えない

日本電気が4月に発売した新型の98，つまりPC-9801UV11/CV21/LV21がいつまでたっても店頭に並ばない。ふと気がつくと，先日まであったはずのVX21もXL2もなくなっている。さらには奇妙なことに，あって当たり前の割引値札がなくなっている……。

これ，メモリ不足で日電の生産が追いつかないため。もちろん日本電気でも生産してはいるが，企業向けや予約客の注文残をこなすのに精いっぱいで，とてもショップに割り当てるには至らない。

さらに状況は悪化しており，いまから予約したとしてもいつ買えるかわからないそうだ。

DECがVAXを値上げ

メモリの調達費用がかさむため，DECはVAXの価格を6月出荷分から5パーセント前後の幅で値上げした。中小パソコンメーカーやワークステーションメーカーも同様の値上げを順次開始している。

危険なメモリが出回る

TIのTTLをDRAMだといって売りさばいた業者が現れたことは有名な話。そこまでいかなくても，不良品や古いパソコンから外したDRAMが出回っている。

とくに人気があるのが，メルコの増設RAMボードから外したメモリ。うまく外したものに当たればラッキーだが，ピンが欠けていたり死んでいたりするものも当然ある。もともと秋葉原のスポット市場にはメーカーはほとんど卸していないので，買えるという話を聞いたらまず危ないと疑うことが必要だという。それでも敢えて手を出すほど不足は深刻なのだ。

休業状態の会社も

米国，大手，大口というのが優先客。こうなるとますます入手できないのが，一品料理を得意とするシステムハウスや周辺機器メーカー，そして中小。したがってこうした企業では，注文を受けても材料がないことには製造できないので，ハード部隊は開店休業状態にしてソフト開発で喰いつないでいる。

まだまだ深刻な影響を受けている話はあるが，こんな感じで食糧難ならぬメモリ難の様相が完全に定着してしまった。ニセ物の横行や再生製品の登場に至っては完全に戦後の闇市的感覚だ。

ボロ儲けの半導体メーカー

メモリがここまで足りなくなった理由だが，おおむね3つ考えられる。

まずは，各メーカーが4年前にメモリを余るほど作ったために過剰在庫になりダンピング販売競争に至ってしまったので，今

したいところに当ててプリントできる。

辞書は固有名詞などを含め約60,000語，イラスト241種類を内蔵している。

最大40文字までを一括変換し，英語など5カ国語に対応する欧文ワープロとしても使用でき，また最大99人分を登録できる住所録管理機能を備えている。

液晶ディスプレイは8文字×2行，内部メモリの記憶容量は，A4サイズ原稿にして約3枚分。

このほか，オプションでイメージリーダ(20,000円)や，48ドットの毛筆と96ドットのイラストパターンが使える毛筆・イラストパック(5,000円),40種類のイラストがハンディプリンタで印刷できるイラストパック(5,000円),外部記憶用メモリカード(3,000円)などが用意されている。

本体プリンタは熱転写式で24×24ドット。本体サイズは幅348×奥行き288×高さ84mm，重量2.2kg。

〈問い合わせ先〉
キヤノン㈱　☎03(348)2121

BOOK

X68000活用研究Ⅲ
電波新聞社

月刊マイコン誌別冊のX68000活用研究シリーズ第3弾。ⅡのX-BASICマスター編を受けたもの。X-BASICのインストール，OSとのインタフェイス，エディタ，構造化プログラミングなどに関する解説。

『X68000活用研究Ⅲ X-BASIC活用 Q&A』
塚越一雄　著
B5判，230ページ，2,000円
〈問い合わせ先〉
電波新聞社　☎03(445)6111

X68000
活用研究Ⅲ

X68000 3Dグラフィックス入門
ビー・エヌ・エヌ

X68000 3D
グラフィックス
入門

X68000を使って画面上に図形を描いてみようという本。XCで記述したオリジナルのグラフィックライブラリを使って，立体図形の表示の仕方や動かし方を説明している。最後の第Ⅲ部では3次元グラフィックを行う上で必要な知識を述べているが，注釈がふんだんにありすぎて少々読みにくい。

『X68000 3Dグラフィックス入門』
ビー・エヌ・エヌ第2企画部編
B5判，212ページ，2,200円
〈問い合わせ先〉
㈱ビー・エヌ・エヌ　☎03(238)1321

回は各社が少なめに作っていること。次に，通産省がダンピングにならないように価格と生産量をチェックしているのでさらに少なめになってしまうこと。そして3つ目は外国のメーカーが4年前に量産競争で敗退したため今はほとんどの外国メーカーでメモリを作っていないこと。このベースになっているのは1986年の9月に日本と米国の間で結ばれた日米半導体協定だ。

つまり「必要な量だけ適正にメモリを作りましょう」ということなのだが，おかしなことには必要最小限なだけしか各メーカーが作らなくなってしまった。

メモリは，需要と供給量で市場価格が決まる「時価」の市況商品だ。したがって品物が多いときには安いが，足りなくなってくると高くなるという性質がある。そうなると各メーカーが生産を抑えれば当然市場で不足してくるし，価格も下がらない。価格が下がらないから量産する必要もでてこない。

いかにも初歩的な経済原則にのっとった商品だが，このせいでメモリの値段はこの1年でほとんど下がっていない。メーカーはボロ儲けし，一方では品物が足りないのでブラックマーケットが横行して犯罪が増えてきた，ということだ。

いま適正価格は256K DRAMが1個350円前後なのだが，時価は実に1,200円もする。それでもまとまった量は手に入らない状態だ。1メガも同様で1個2,000円のものが4,000円以上する。1メガなど本来は1個1,500円以下まで安くなっていて当然なのだ。それだけ不当にメーカーは儲けていることを示している。

いつまで続くか？

このようなおかしな状態がいつまで続くかが肝心なところだが，各種予測によると現時点に比べて年末時点では1M DRAMは7割から8割ほど多めに生産される。256K DRAMは逆に1割から2割ほど減る。これで需要が満たせるかどうかがポイントだ。

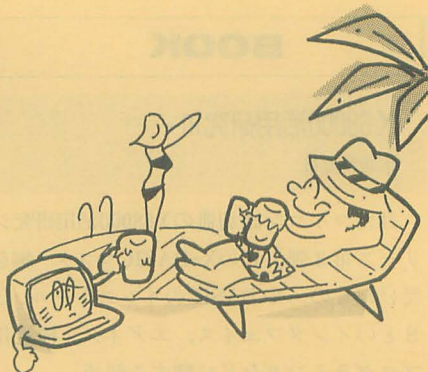
結論からいえば，状況としては改善されるものの相変わらずメモリ不足は解消しないだろう。ブラックマーケットは一向に減らないようだし，価格もそう安くはならないずだ。

まあ，現在のようなおかしな状態は減ってくるだろうから秋には98もちゃんと買えるようになる程度には回復するだろう。とはいっても OS/2も出てくるし，ワークステーションやパソコンで標準メモリ容量が4Mバイト，5Mバイトとグングン増えているところをみると，ひょっとしたらさらに不足が激しくなるかもしれないという懸念も出てくる。

なお，今回のメモリ恐慌でいちばん喜んでいるのはPC-9801関係者かもしれない。

というのは互換機，対抗機がどんどん出てきて相当な苦戦を余儀なくされているはずだったのが，しばらくはVX で殿様商売ができているのだから。このメモリ不足による恐慌状態は確実に98の寿命を1年延ばしたことになる。

今月はコンピュータ関係で目立った動きがなかったので，このようにメモリの話でお茶をにごしておくことにした。とはいってもこのような状況もなかなか興味深いと思う。次回はちゃんとコンピュータの時評を書く。

こうした話も一応は頭に入れておいたほうがいいので，ヒマな人はいちど256K DRAMを買うふりをして，秋葉原にでも遊びに出かけてみるのもいいだろう。　(K.T.)

FROM READERS TO THE EDITOR

いよいよ夏本番となってきました。夏休みも，もう目前。うっとうしい梅雨の間部屋に閉じ込められていたウサを晴らしに海や山に出かけて，思いっきり羽を伸ばして遊んじゃいましょう。でも，学生の方は宿題があることを忘れないよーに。

◆6月号の特集「恐怖のプロッタ＆スキャナ攻撃」はとても楽しく読めました。今度特集でプリンタの活用法なんかやってくれたらうれしいなー。それもビデオプリンタみたいな最新のものじゃなくて，昔からあるドットプリンタやプロッタプリンタなんかの面白い使い方をやってほしいものです。　堀内　雅貴（15）北海道

　あのような活用法が見つかれば，またご紹介していきたいですね。でも簡単に見つかるほど材料はコロがっているのかな。ほかにもいい活用法があれば堀内君も考えてみてくださいね。

◆6月号の特集は，なかなか奥が深いようで，これから私もじっくり考えていきたいと思います。それと田村さん，うちのmodel 10も元気です。あと，うちの学校には桜の木が26本もあるので少々怖かったりします。

清水頭　武信（16）青森県

◆あぶない福袋にはマイッタ。うっかり全部信じるところでした。　対馬　恵一（33）青森県

◆特別記念番組「これ，なんですか。」はなかなかよかった。しかし，番組中にドラマがないのが残念でならない。次回はドラマもお願いします。それにしても超長寿番組「砂の嵐」はよかった。　鹿沼　一洋（18）栃木県

　鹿沼君みたいに，あの「砂の嵐」を笑って読んでくれればいいんだけど，意外と「砂の嵐ってどこに載ってたんですか？」っていうハガキが届いていたりするのですよ。

◆「あぶない福袋」のX1/X1turbo用Z'sSTAFF 68Kというのを見て，これは本当かな，と最後まで読んだらガッカリした。誰か本当に作ってくれー。　板垣　一彦（16）北海道

　福岡の高橋哲史君って，有名人だったのね，こういう意味では。

◆あの“あい○っと”君にはまったく参ってしまいました。だって，僕はICOTの研究員なんだもん。F所長にいいつけてやる。

和田　正寛（25）奈良県

◆6月号の78ページ「編集室の逆襲」で，僕へのコメントは僕の前の人へのコメントではないのですか。人の名前を勝手に間違えないでください。　玉木　俊秀（20）鳥取県

　ごめんなさい，こちらのミスで玉木さんと東京都の村井裕弥さんとを間違えてしまいました。玉木さんからのコメントはというと，「ソニーのトリニトロン管を使ったディスプレイテレビをシャープから出して欲しい」というものでしたね。ご迷惑をおかけしましたので，改めてここに返事を書かせていただきます。「とんでもないヤツ」

◆5月20日（広島県でのOh!X発売日）に，テレビOh!XでAM7:00からおかしな番組を放送していました。本当におかしな番組だったから，思わず放送終了まで見てしまい，ついでにビデオにまで撮ってしまいました（冗談）。

住田　永司（16）広島県

◆権兵衛頓馬伝と違って，わが町の農協のサイレンは9時，12時，5時に鳴ります。ちなみに正午のサイレンと一緒に有線放送で農民体操が始まるのさっ。その内容とは「さあ皆さん，農民体操を始めましょー。（中略）なおこの体操は朝昼晩3回欠かさずにやるよーにしましょー。農作業の合間にもやりましょー」というものです。　松永　和久（18）熊本県

　実際は，どこもサイレンと一緒にこういうオマケが付いているものなんですか。ハガキを送っていただいた大野真美さんに，バージョンアップのときはこのように強力な農民体操も入れてもらうようにお願いしちゃいましょ。

◆Oh!Xのテーマをテープに録音してエンドレスで流していたら，あの曲が耳に焼き付いてしまって，夜眠れなくなってしまった。それにしてもみんなで口ずさんで広めるには難しい曲ですね。　渡辺　光（15）北海道

　口ずさむにはいちばん最後のところがネックかもしれません。それにしてもエンドレスで聞いたというのは凄い。

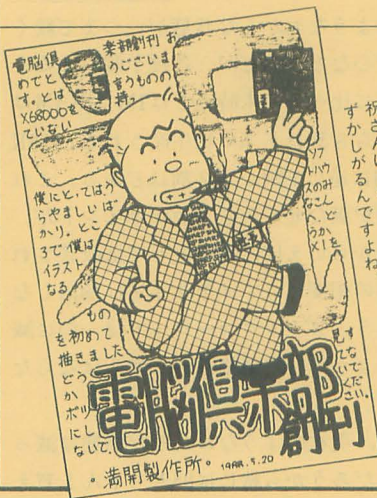
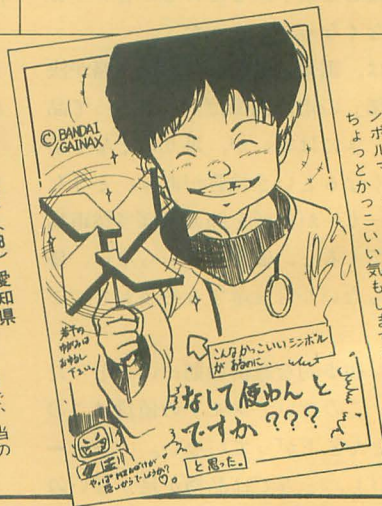
◆「ピー，カンマが付いていません」，「ピー，文末の記述が誤っています」。「へえ，私が悪うございました」と，かわいいX68000にBASICを教えていただいているこの私です。あー，変数，変数，変数，もおやだよー。でも，エラーメッセージが日本語だから僕みたいな初心者にもわかりやすいよね。あと何年かたったら「ピッピー，あなたの指示は信頼できません。あとは自分でやります」なんてメッセージが出るようになったりして……。いま地図帳を作っています。あともう少しがんばるぞー。

吉田　裕（16）岩手県

◆X1turboZのローンが終わらないうちにX68000ACE-HDを購入しました。いまではバブルボブルのBGMとともにOSが立ち上がり「おはようございます」と起こしてくれます。BEEP音はというと，「こんな計算私にやれっていうんですか？」と森川美穂の声で話しかけてくれます。そのうちX68000と掛け合い漫才でも始めようかなあと考えています。

深井　克志（26）長野県

　吉田君や深井さんがX68000と接しているときの姿が目に浮かぶようです。でも，漫才しながらプログラムが組めるようになれば，きっと最高だろうなぁ。

◆ビジネスショウでOS-9/X68000を見た。負けるなHumanというものの，OS-9は私の考えていたものより数段素晴しかった。実際にあのま

◀中西　弘泰（18）愛知県
これは豪快なイラストですね。ところで，当の祝さんは「社長～」なんてひやかすと本気で恥ずかしがるんですよね。

◀福原　徹　埼玉県
ぎょえ～！　さんぜんと輝くXファミリーのシンボルマークじゃあ。しかし，ここまでやるとちょっとかっこいい気もしますなあ。

ま商品化されれば最高だ。あのパーソナルウィンドウのデモがよかったな。X68000がLANで動くなんて……。ふっふっふ，商品化されればOS-9で制御をやるのじゃあー。

清水　克俊（26）宮城県

来月の9月号で，OS-9/X68000の追跡レポートをお届けする予定なので，お楽しみに。

◆最近，雑誌などでよく「台湾」を取り上げた記事を目にしますが，もし編集室の皆さんが来台されることがありましたら，台北の秋葉原もアニメイトもよく知っているこの私に，ぜひご連絡ください。

高綱　慎二（31）中華民国台湾省

◆あのー，いつか特集で「私はこうして大学に合格した」とか，「こうすれば大学に合格できる」などとゆーのをやってもらえないでしょうか。

小松原　秀貴（17）千葉県

どーも，編集室にいる人材を投入してこのテの話題となると「自慢じゃないが，私はこうして3浪した」とか，「ストレートに入って，2年留年しても威張っていられる方法」とかっていう特集のタイトルになってしまいそうなんだけど，小松原君はこんな内容の特集でもいいのかなぁ。

◆6月号で一番よかった記事。164ページに出ていた大阪の遠藤さんのハガキ。

田原　孝（16）山口県

結構多かったんだよね，田原君みたいに遠藤さんの意見が最高だったと言ってきた人が，そこで皆さんからのリクエストにお応えして，再び遠藤さんにご登場していただくことにしましょう。

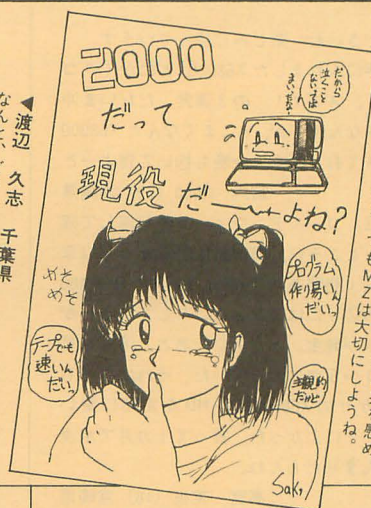
◆6月号で「68,000円でX68000を買ったと嫁さんをだました」と紹介された遠藤です。ナンちゅう自己紹介や。過去これまで2度ほどハガキを取り上げていただいていますが，いずれも嫁さんネタでした。前のは「X1のゲームよりファミコンのほうが面白いと嫁さんがいうので大阪湾に沈めてやろうか」などという内容のもので，この前のハガキと合わせてずいぶんと嫁さんをいじめてしまいました。しかし，本当は優しい嫁さんです。結局，皿洗いも1カ月で許してもらいました。そこはそれ，やっぱ夫婦でんがな，ホレホレ(失礼しました)。しかし，時々98を見ては「あれ9,800円」とか，88が「8,800円安いわー」などと未だにいわれています。

遠藤　勇（31）大阪府

◆私も1年ほど前のある日，奥さんに「X68000買ってくれ」といったところ，「自分の小遣いで買えば」といわれてしまった。そこで，えーいままよ，とばかりに小遣いのなかからローンを組んでX68000を買ってしまったが，それからはタバコ代にもことかく始末。しかし，うちの奥さんはいい奥さんです。タバコ代がなくなったので「ヤニがきれた，ヤニがきれたー！」と，畳の上でもがいていたら，タバコを買ってきてくれました。今度は「ソフトがきれたー」ともがいてやろうかしら……。

渡辺　信一（29）岩手県

▲渡辺　久志　千葉県
なんと、ビジュアルシェルならぬ「美女あるシエル」とは。ちなみに私（S.S.ですヨロシク）は、RAMDISK．SYSを指名したのですが。

▲佐久間　隆弘（18）千葉県
さあ、涙をふいてキーボードをたたいてごらん。なあんて、いくら絵が可愛くても私は男を慰める気はないのだ。でもMZは大切にしようね。

ほんとに，遠藤さんも渡辺さんもいろいろとよくやりますね，奥さん相手に。このお2人を見ていると，世の中の亭主族はより仲間意識が強まってしまうんでしょうね，きっと。でも，このあとの奥様族のご意見を見るとあんまり笑ってばかりいられなくなりそうですよ。

◆はじめまして，編集室の皆さん。今年の3月21日に結婚したばかりの新妻です。婚約中，ダンナ様は3年前に購入したばかりのPC-8801mkⅡSRを山ほどあったソフトと一緒に手放しました。そのときは「もう結婚するんだから……」と，泣かせのひと言。私の実家の母は88がベラボーに高いのを知っていましたから「なかなかしっかりした，エエお人や」と感心することしきり。そして現在，ナゼか我が新居の狭いアパートにはX68000ACE-HDが“でぇーん”と居座っています。そしてダンナ様はほとんど毎日，源平討魔伝の安駄婆に叱咤されています「オロカモノメッ」と。……ったく，88をソフトごと手放すわけよね。ヒトの持参金でこんなモン買いやがって。おまけに夏のボーナスでプリンタも買うんだとか。おかげで遊びに来た実家の母を玄関先で追い返す始末。どーしてくれるんだっ！　と文句をいいながらもX68000用のカバーを手作りしているあたりが，やっぱり新婚っていうのでしょうか……ネッ。

畠中　三代子（25）東京都

そうか，嫁さんの持参金という手があるわけか。いやー，独身者の多い編集室にとってはこのテのハガキは目の毒だけど，結婚についてのいい勉強になりました。ハイ。

◆私，X68000が欲しいんです。ワープロ専用機歴は趣味と実益で意外と長いんですけど，コンピュータは10年ほど前に高校の情報処理の授業で紙テープにパンチするといった作業の経験だけで，パソコンなんて耳学問と目学問のみです。それだけにどうしてもX68000が欲しくて，欲しくて……。最近，男性の皆さんは奥様攻略に苦心なさっているようですけど，うちではファミコンとPCエンジンさえあれば生きていけるという夫を説得するのがたいへん。どうやって夫をだまくらかしてX68000を購入すればいいのか，ダンナ族の弱点を誰か教えてください。ちなみに昔，車を買うときは「これで送り迎えしてあげる」といってだましました。でも一度もしたことがないんで，とうとう車は売り飛ばされてしまったんです……グッスン。

奥山　享子（28）大阪府

ご主人は結構ゲームがお好きなようですから，ここはひとつ店頭にズリズリと引きずって行って，源平かスペハリのデモなんかをさりげなく見せてひと言，「これで値段は68,000円！」……これじゃ遠藤さんの二の舞か。

◆妻からひと言：X68000を購入してからもうすぐ1年になります。メカオンチの私は，当初は憎悪の目を向け，かつ，それに熱中する主人には無言の抵抗を続けたのです。しかし，この5月，スランプに陥る主人をしり目に，密かにBASICなるものを勉強し始めました。そしていつの日か，主人に教えてあげるのが目標です。がんばるぞー。だからやさしくて，実用的なプログラムをたくさんのせてネ。

主人からひと言：どおりで知らないうちにディスクがファイルで一杯になっていたわけだ。

村上　博文（38）兵庫県

ご主人のスランプの隙を突く，なんていうところが，しっかりご夫婦してますね。でも，こうして2人で覚えていけばきっと上達も早いのでは。

◆さて，皆さん。1986年2月号のOh！MZを用意してください。そしておもむろに132ページを開いてください。そのページは「第1回言わせてくれなくちゃだワ」の北海道地区ですね。そして2番目の人の文章を見てください。低レベルの意見ですね。持っている機種はというとファミコンと書いてあります。恥ずかしいですね。最後に名前を見てください。「大渕正人」と書いてありますね。それは私です(どっとはらい)。追伸，私は1984年7月号からOh！MZ(X)を買っていますが未だにナイコン(死語)である。

大渕　正人（18）北海道

でも，大渕君も秋にはX68000を購入予定とか。そのときはハガキの「あなたの愛機は？」というところにデッカイ○印を付け

てくださいね。楽しみに待っています。

◆ついに手に入れましたX68000。とにかくスゴイ，スゴイ，スゴイ！　の3連発。ただいま沖縄に出張中なんですが，そこまでなんとX68000を持ってきております。今晩も抱いて寝よーと。

宮内　功和（29）沖縄県

◆5月3日，ゴールデンウィークを利用して帰省していたところに，熊本気象台始まって以来の降雨量（1時間あたり140ミリ）で，標高400メートルの辺りには滝がゴロゴロしているわが町では洪水が発生。水が引いたあと，滝壺を見ると車が4～5台はまっていた。それにしても買ったばかりのX68000ACE-HDを宮崎に置いてきてホントによかった。買って1カ月で水浸しじゃあんまりだもんね。

高宮　克也（18）宮崎県

◆6月号77ページの「NEW BASICコンパイラ」を見て，「アレッ，まだ出ていなかったんだっけ？」と大ボケをしたのは私だけでしょうか。これには期待していたので，とても懐かしいような気がしてしまいました。

若月　一弥（18）新潟県

◆6月号133ページの欄外に載っていた榎木さん，私も中島みゆきのファンで，アルバムは友だちと協力してすべて持っています。ついでに谷山浩子と戸川純と富田靖子と小泉今日子とBaBeのファンでもあります。こんな私を友人たちは「趣味のわからんやつ」といっています。おっと渡辺美里とプリンセス²を忘れるところでした。先日，やっと念願のCDプレイヤーを買いました。これで歌麿が聴ける！

堀端　英彰（19）東京都

◆榎木さん，ここにもいますよー。

藤原　将騎（19）愛知県

Oh！Xの読者のなかには中島みゆきファンって多いようですよ。このほかにも，同じような榎木さん宛のハガキがたくさん届いていましたから。

◆いま岐阜県の中学校数学会は，コンピュータの導入を考えています。しかし，BASICなどの指導者が不足しているようです。ムハハハ，ようやくわしの時代がやってきたー。

岩腰　清（33）岐阜県

岩腰さんって，当然，中学の先生なんでしょ。こんなに明るい先生の数学の授業って面白そう。

◆ALANを1週間かけて打ち込んだ。これは，はっきりいってスゴイ。まさか視点の移動までするとは思わなかった。でも，各面ごとにボスキャラはいたほうがいいし，タイトルももっと派手なほうがよかったような気がする。それから最後のボスキャラは祝一平氏だというウワサですが，本当なのですか。

中野　貴大（15）埼玉県

◆6月号でX68000用ゲームプログラム「信州」を発表させていただいた飯島です。信州の発表に関しては，数多くの方にご協力いただいたようで，たいへん感謝しております。さて，ここで掲載されていなかった隠し技をお教えしておきます。本文中には「チャイはできない」と書かれていますが，実はマウスの右ボタンを押しながら左ボタンを2回クリックするとその面の最初に戻れるという，私だけが知っている必殺技が隠されていたのです。

飯島　匡史（21）長野県

飯島さんの「信州」には，たくさんの方から面白いというハガキをいただきました。それに，もうこの裏技を見つけたというのも確かあったような気が……。いずれにしても飯島さんどうもありがとう。次回作にも期待しているからね。

◆Mr.プロ野球はとってもいいですよ。でもポケットに手を入れて抗議に出ていけないのがとっても残念です。　渡辺　俊彦（15）栃木県

◆SFTOUCHの西川善司さんへ。源平で虎を出さないようにするのは，虎が出す銭の王を取らなければ出なくなるはずです。

五藤　章博（18）徳島県

◆ある日，源平討魔伝をやっていて突然，思った。「うはははは，たわむれは終わりぢゃ」という頼朝の顔を郷ひろみの顔に変えて，「うはははは，郷ひろみです」と言わせると面白いかもしれない，と。　白石　達也（18）福岡県

◆えー，これまで発表してよいのやら迷っていたのですが，X68000の弱点をあえて発表することにします。その弱点とは「引っ越しのときに箱にしまいにくい」という事実です。これは実際に経験してわかりました。以上，報告終わり。

佐藤　哲哉（24）東京都

それはたいへん。さっそくシャープさんにこのことを連絡してあげなきゃ……，ンなわけないでしょ。

◆編集室のU氏へ。1)変身忍者嵐の愛馬は「ハヤブサオー」，2)ダイアモンドアイの光線は「外道照身霊波光線」，3)バトルホークの必殺技は「戦刃旋風斬り」，「変幻風刃投げ」，「必殺三方刃」などなど。4)のとんがり帽子メモルについてはアニメのメカものしか興味がないのでパスです。でも相当マニアックな問題でしたね。では，以下の問題に答えてください。1)白獅子仮面の変身のときの叫び声。2)緊急指令10-4・10-4のチーム顧問の名は。3)突撃ヒューマンの主人公「岩城潤一郎」の出身地は。4)電撃ストラダ5の国際的シンジケートの名前は。以上，マイナー作品ばかりですが，ぜひ答えてください。

砂田　陽一（20）鳥取県

せっかくの砂田さんからの挑戦状も，U氏は今月多忙のため解答をもらうまでに至りませんでした。どなたか自信のある方は編集室までご返事ください。

◆北海道では「ミスター味っ子」が朝6時45分からやっているので，見るのがチトつらかったりします。　福本　雅一（17）北海道

◆先日，映画館で「うる星やつら全5作＋めぞん一刻」をオールナイトで観た。オールナイトなんだから，どーせ観客は少ないだろうと思っていたら，さにあらず。場内は懲りない面々で満員になってしまい，座れない人のためには折りたたみの椅子が用意されるという始末。映画は午後9時30分から朝の6時40分まで上映されたのですが，私は一睡もしないで全部観てしまいました。　望月　隆（23）東京都

◆ある日，京都のアバンティに行ったとき，シャープの電子手帳が置いてあったので，書き込まれていたメモを見ていたら，なんと「Oh！Xはローディストだ」と書いてあった。僕は目が点になった。　吉田　茂樹（20）奈良県

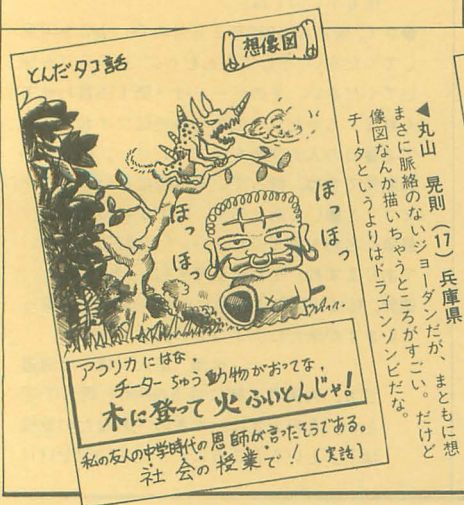
困ってしまいますね。こんなところに，こんなことを書かれてしまって。最近，よくOh！Xはオタクだとかローディストだとか言われているようですが，まったくアレは迷惑な話です。正確には一部の人間が実際にそうなだけで，決してOh！Xではない……，イカン，これじゃフォローになってない。

◆うちの嫁さんは，X1twinの「これがX1誕生の5年目の解答です」という広告を見て，「ねぇ，5年前に出された問題ってなぁに？」と，わけのわからんことを言っていました。

中野　春一（27）東京都

◆私は毎日イライラが続いています。なぜなら昨年の10月に2000ccの車を買い，次にケンウッドのミニコンポを買い，またそのすぐあとにX68000を買った。さらには中古の250ccのバイクを買い，そしてとどめに新車のトゥデイまで

◀丸山　晃則（17）兵庫県　まさに脈絡のないジョーダンだが，まともに想像図なんか描いちゃうところがすごい。だけどチータというよりはドラゴンゾンビだな。

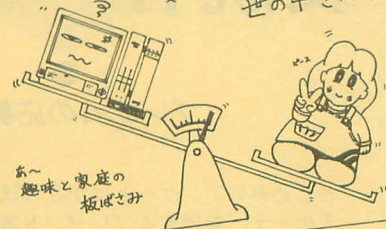
◀山崎　潤一（19）福島県　2年連続イラスト大賞（ちゃだワ参照）の山崎君のCG初登場。ところで高橋君から連絡があったらちゃんと勉強するよう言ってください。

買ってしまった。実はこれはみんな妻には黙って買ったものばかり。私の衝動買いはもうほとんどビョーキだ。いつかバレるのではないかと毎日ヒヤヒヤ。そして残金の支払いと妻の怖い目にオロオロ。えーい，こうなりゃ自分とこの店の経費でみーんなオトソーっと!!

田中　伸和（30）大阪府

これだけのものを勢いだけで買ってりゃ，立派に病気ですよ。でも，X68000 1台買って奥さんにいじめられてる世の亭主族からすれば，こんなにいっぱい買って経費で落とそうなんて考えてる自営業の田中さんは，きっと怨まれちゃうだろうなー。

◆シャープはMIDIボードを早く出さないのだろうか。音楽のジャンルでPCやMacに1歩も2歩も10歩も100歩も後れを取っている感じである。せっかくX68000にはFM音源ボードやADPCMを載っけたりしているのだから，MIDIも出すべきである。こうなりゃ，やはり自分で作るかよその機種ででも買うしかないのだろうか。さあ，どうーすんべ。

柳谷　敏之（18）神奈川県

ホント，MIDIボードに関しては我々シャープユーザーは，今月のOh！Xでもご紹介しているように自作するしか方法は残されていないんでしょうか。せめて，年内には発売してほしいものですね。みんな待ってるんですから。

▲祐成　好規　東京都

こりゃ目方でドンの世界だね。X68000ってセットで20kg程度だから……っと，そーゆー問題じゃないか。やっぱり奥さんは大事にしてあげないとね。

ぼくらの掲示板

●掲載ご希望の方は，官製ハガキに項目(売る・買う・氏名・年齢・連絡方法……)を明記してお申し込みください。
●ソフトの売買，交換については，いっさい掲載できません。
●取り引きについては当編集室では責任を負いかねます。
●応募者多数の場合，掲載できない場合もあります。

仲　間

★MZ-2500，X1turboユーザーズクラブ「T.A.C」では，第2次会員を募集します。活動内容は毎月1回発行しているディスクによる会報を中心に，いろいろなことを行っています。入会希望者は60円切手同封のうえ封書にて連絡を。〒569 大阪府高槻市大塚町2-55-4　北浦真一（18）

★「Team X's BLAZE」ではX1turbo/X68000ユーザーの会員を募集します。活動は主にディスク会報を発行のほか，ソフトの共同開発や情報交換などです。興味のある方は往復ハガキで連絡を。〒533 大阪府東淀川区大桐4-5-24　小倉大次郎

★「ZIP-X1」ではX1/X68000ユーザーを中心とした第2期会員を募集します。活動は会報発行を中心に，現在はAVGの共同開発に取り組んでいます。ですからマシン語に詳しい方やCGに自信のある方は大歓迎。初心者の方にはシナリオなどで参加していただければと思っています。また，投稿プログラムの年間ポイント制による豪華プレゼント大会なども実施しています。興味のある方は切手300円分同封のうえ封書にて連絡を。〒037-03 青森県北津軽郡中里町豊島　工藤陵（17）

★X1/X1turboユーザーズクラブ「FREE X1」では，第1次会員を募集します。活動はゲームの情報交換を中心に行いますので，とにかくゲームが大好きだという方，大歓迎。入会金，会費，会報代など一切無料です。詳しいことはご意見やゲーム歴などと一緒に，60円切手同封のうえ封書にて連絡を。〒633-02 奈良県宇陀郡榛原町天満台東3-14-3　土屋信（18）

★X1/X1turboのディスクユーザーを対象としたクラブ「うれし〜」では会員を募集します。ゲームの好きな方，ぜひご参加ください。入会金不用。連絡は60円切手同封のうえ封書でお願いします。〒028-05 岩手県遠野市早瀬町2-4-11　松田孝幸（16）

★6月号にX1turbo model10ユーザーの話が載っていましたが，私のmodel10は拡張に拡張を重ね，見た人を驚かせるまでになっています。G-RAMの増設や5インチFDDだけではもの足りずCZ-300Fの3インチを取り払って，電動ポップアップの3.5インチ2DDをデュアルでつなぎ，FM音源も搭載して，CP/M上でTURBO PASCALやHI-TECH Cを使うなど，かなり金をつぎ込んでしまい，手放す気にはなれません。Oh！Xの読者の方で同じように使っておられる方がいらっしゃいましたらご連絡をください。お待ちしてます。〒236 神奈川県横浜市金沢区並木1-5-410　岡本努

売ります

★データレコーダCZ-8RL1を1万円で。連絡は往復ハガキで。〒125 東京都葛飾区柴又3-36-1-103　青木賢一

★FM音源ボードCZ-8BS1を1万4千円，ミュートピアCZ-139SFを4千〜6千円，またX1用マウスを6千〜9千円で。いずれも完動，付属品一式付き，送料別。連絡は往復ハガキで。〒004 北海道札幌市豊平区平岡2条1-57-14　八木明（38）

★FM音源ボードCZ-8BS1（ソフト付き）を1万円，X1用カラーイメージボードCZ-8BV1（turbo用ソフト付き）を1万5千円，プリンタMZ-1P17（X1用ケーブルとインクリボン10本付き）を3万円，ポケコンPC-1360K＋CE-140PK＋CE-2H32M(新同)を4万円，PC-1261＋CE-125(取説なし)を8千円で。すべて送料込み。連絡は往復ハガキで。〒707 岡山県英田郡美作町上相852　小林英樹（24）

★アイワのモデムPV-A1200mkを1万6千円，FM音源ボードCZ-8BS1を1万3千円，2つセットの場合は2万8千円で。いずれも箱，マニュアル，付属品付き。またOh！MZのバックナンバー1985年11月号以降のものを1冊1,000円で。連絡は往復ハガキで。〒305 茨城県つくば市観音台1-7-2　近森敬一（17）

★X1用漢字ROM・CZ-8BK2を箱・マニュアル付き，送料込み1万円で。連絡は往復ハガキで。〒982 宮城県仙台市郡山新々田東6-166　岩淵剛（18）

買います

★プリンタCZ-8PC1を2万5千円で。連絡は往復ハガキで。〒468 愛知県名古屋市昭和区天白町八事裏山60-48第2向陽荘　小山徳章

★MZ-80B用MZ-2000コンパチボード(セキグチ)SID1002＋MZ-2000用G-RAM(1，2，3)を送料込み1万〜1万5千円で。連絡は往復ハガキで。〒245 神奈川県横浜市戸塚区深谷町25深谷団地10-55　黒武者健一（18）

★X1用漢字ROM・CZ-8BK2を8千〜1万円で。箱なし可。連絡は往復ハガキで。〒779-01 徳島県板野郡板野町松谷シシトキ南3　和田孝史(15)

★FM音源ボードCZ-8BS1(付属品付き)を1万2千円で。付属品のない場合は1万円で。連絡は往復ハガキで。〒861-04 熊本県鹿本郡菊鹿町木野319　井手上智一（16）

バックナンバー

★Oh！HCの第8号より最終号までを各500円で。連絡は往復ハガキで。〒489 愛知県瀬戸町小田妻町1-273-3-402　岡本一（26）

★Oh！MZ1984年8月号を送料込み1,500円で。ハガキ以外の切り抜き不可。連絡は往復ハガキで。〒275 千葉県習志野市袖ケ浦4-6-6　原英樹

愛読者プレゼント

プレゼントの応募方法

とじ込みのアンケートはがきの該当項目をすべてご記入のうえ，希望するプレゼント番号をはがき右下のスペースにひとつ記入してお申し込みください。締め切りは1988年８月18日の到着分までとします。当選者の発表は1988年10月号で行います。

2

シャノアール ☎03(702)0598

プロフェッショナル麻雀悟空

X1シリーズ用
5"2D版 6,800円

3名

腕前に合わせて24人のなかから対戦相手を選べる本格派麻雀ゲーム。研究モードや段位戦，勝ち抜き戦などもあり，対局データもいつでも見られる。

1

データウエスト ☎06(968)1236

第4のユニット2

X1/X1turbo用
5"2D版3枚組 7,600円

2名

４月号のゲーム特集でもご紹介した第４のユニットに，新たな強敵が加わって登場のバージョンアップ版。アドベンチャーアクションの世界を堪能してください。

3

ソーサリアン スーパー・アレンジバージョン

CD 3,200円 3名

ソーサリアンのオリジナルサントラ。ゲームに挿入されているもののうち13曲を選んでアレンジしている。全曲楽譜付き。

6月号プレゼント当選者

1 ザ・ラスベガス（広島県）東有明（東京都）稲田昭（北海道）東理義明 2 WINDEX PRO-68K（宮崎県）田井秀樹（福岡県）松尾和茂（北海道）荒瀬匡宗 3 Z'sSTAFF PRO-68K（大阪府）田中賢治 4 魔神宮（三重県）小村定（北海道）青山利之（秋田県）簗瀬信悦 5 T.D.F.（山口県）安達信義（神奈川県）原英樹（東京都）山岡義道 6 桃太郎伝説（千葉県）新井政樹（愛知県）会田雄一郎（埼玉県）小山洋一 他７名 7 XLink68（東京都）佐々木孝司 8 EW（栃木県）秋山吉康（千葉県）小宮山政敏（静岡県）秋山誠 他３名 9 C-TRACE68（神奈川県）清水雅夫 10 FINAL（千葉県）人吉一馬（鹿児島県）洞克治 11 源平討魔伝（福岡県）中山高幸（北海道）堂前敏弘（山口県）岸勝利 12 いろはにほへと X68000用（神奈川県）稲葉敦司 X1用（大阪府）松村一郎 13 おさな妻奮戦記（香川県）伊藤浩克 14 億万長者（大阪府）勇尾繁輝（東京都）廣澤亮輔（神奈川県）嶋田哲朗 15 超戦士ザイダーグッズ a.（沖縄県）伊舎堂盛行（大阪府）藤本修一 他18名 b.（東京都）川崎美奈子（千葉県）浦川博之 他18名 c.（新潟県）上田勇（京都府）嶋田睦雄 他18名 16 レジェンド（長野県）宮田淳（神奈川県）今里吉伸（大阪府）藤田義弘 17 ジプシ（岐阜県）橋本広治（熊本県）福田健児 18 ガイフレーム（京都府）伴哲也 19 ロゴシール（群馬県）馬場文明（千葉県）田中信弘 他14名 20 麻雀狂時代SPECIAL（東京都）梅津紘（埼玉県）青木高博 21 ストーム（富山県）金戸俊道（北海道）藤野恵子 22 ディーダッシュ（千葉県）秋葉貴男（兵庫県）宮本康司 23 a．キーホルダー（山梨県）望月洋紀（福島県）田村泰司 他３名 b．メモ帳（２冊組）（愛媛県）近藤崇（宮城県）緑川健 他３名 24 ぎゅわんぶらあ自己中心派（新潟県）佐々木憲一（熊本県）永田紳治（東京都）比留間秀哉 25 王子ビンビン物語（栃木県）藤井弘（埼玉県）田村佳則（愛媛県）松本篤実 26 ワールドイングス169 a．X1turbo用（千葉県）井村信二（神奈川県）大村邦嘉 b．X1turboZ用（大分県）小野康夫（石川県）奥本広 27 棋太平（福岡県）野中秀一（愛知県）色部弘之 28 リ・バース（滋賀県）湯沢のり子（福岡県）大津和之 29 クリムゾン（千葉県）田村雅樹（神奈川県）市間健太郎（茨城県）増本善久 他２名 30 ミスタープロ野球（京都府）寺内正記（茨城県）木村和弘（神奈川県）野村信重 他２名 31 今夜も朝までPOWERFULまあじゃん（神奈川県）星野俊英（大阪府）松本悟朗（北海道）熊岡忍 32 紫醜羅（埼玉県）秋山豊（宮城県）山本潮（長野県）浜田成裕 33 ドームグッズ a．ポスター（大阪府）八木一桐（神奈川県）山島宏紀 他４名 b．ミュージックテープ（群馬県）新井修一（東京都）米山公一郎 34 テレホンカード（北海道）木下博嗣（宮城県）藤沼俊幸 他３名 35 ディスクケース（兵庫県）橋本弘章（福岡県）高橋哲史（石川県）橘俊次 他３名 36 スーパーレイドックTシャツ a．（東京都）小山宏美（岡山県）池田訓章（栃木県）玉木康夫 他７名 b．（宮城県）山本幸一（鳥取県）松尾論（神奈川県）中嶋祥史 他７名 c．（岡山県）寺尾文治（広島県）島田栄三 他３名 d．（神奈川県）金子満生（北海道）工藤光吉 他３名 37 メモ帳（福岡県）末次正浩（神奈川県）上地剛（山形県）宗片陽一 38 オリジナル下敷き（山形県）武藤俊哉（大阪府）豊嶋則雄（岩手県）高橋伸英 他47名 39 ソーサリアングッズ a．（三重県）坂直樹（東京都）仲本英生（和歌山県）堀内直明 他17名 b．（宮城県）亀卦川宏人（奈良県）平木朝一（静岡県）小杉文武 他７名 c．（東京都）本木康雄（群馬県）平賀修（山梨県）奥山信弘 他７名 40 SUPER大戦略ゲームカード（長崎県）中村浩司（広島県）安部広朗 他３名

（敬称略）

以上の方々が当選されました。おめでとうございます。品物は順次発送いたしますが，入荷状況などにより遅れることもあります。また，公正取引委員会の告示により，このプレゼントに当選された方は，この号の他の懸賞には当選できない場合がありますので，ご了承ください。

編集室から from E・D・I・T・O・R

DRIVE ON

このコーナーでは，本誌年間モニタの方々のご意見を紹介しています。今月は6月号の記事に関するレポートです。

●以前は私にとってのシステム環境といえばBASICでしたから，使いやすいコマンドの多いBASICを積んだマシンがもっとも優れていると思っていました。現在では，やはり良いエディタと，良いFEPと，使いやすいアプリケーションのあるOSが優れたシステム環境ではないだろうか，と思っています。つい最近までアセンブラはアブソリュートアセンブラとしか考えられませんでした。DUAD-88は見たことがある程度ですが，EDASM(ZEDAも似ている)よりも使いやすいとは思えません。それに，S-OSのコストパフォーマンスは高いと思うし，ZEDAで決まりでしょう。それに，華門氏の言うように，恵まれた環境よりもそうでないところでプログラミングを始めたほうが，より血肉になると私も思います。
岡田　忠宏（19）MZ-2200，X68000　広島県

●パソコンのシステムではできるだけ情報を公開してほしい。私の持っているポケコンの話になるが，システム・サブルーチンの一覧表があるにはあっても入出力レジスタすら書かれていなかった。「システムを読むためのアセンブラ入門」を読んで，たしかにソースの解析は勉強になり知識もたくさん得られると思った。しかし，以前あるワープロソフトのかな漢字変換プログラムを逆アセンブルして死にそうになったのを思い出してしまった。初心者はこういうときは気をつけなければいけませんね。
関根　孝司（20）MZ-1500，PC-1480U　東京都

●マシン語は，まず自分で作ってみて，うまくいかない→他人のプログラムを見てみよう→あっ！　そうか！　とやっていくのが一般的だと思います。失敗や困った経験が大切なのです。それから「あぶない福袋」のSPECIAL HARRIER は最高に笑えました。しかし，パロディとはいえ，よく考えるとみんな読者の望んでいることなんですね。
福留　英明（18）MZ-2500　東京都

●システムはユーザーにとってツールボックスでなければならない，と私は思います。単一の製品であってはならないのです。ユーザーが好きなときに，必要な部品をその中から取り出して利用できなければいけません。とにかくいっぱい道具が入っていてその1つひとつが本当に有効で，しかも何にでも使える。また，それは拡張や切り離しが自在で，わがままなユーザーの要求に常に柔軟に応えてくれる。優れたシステムとは，まあ，こんなところだと思います。「よりよいソフトウェア環境のために」についてですが，パソコンを今のワープロとこのように比較するのはちょっと無理だと思います。それは，パソコンには汎用性を持ち続ける必要があるからです。何にでも対応しなければならないから，起動直後にプログラムの入力を素人に求めるような形にもなるのでしょう。我々は，多摩氏の言うように，ハード，ソフトの製品に関するフィードバックを，メーカーにぶつけ続けなければならないと思います。
原　悟（18）X1turboII　宮城県

●私がMZ-1200でマシン語のプログラムを組むのに使用していたアセンブラは，SP-210-2104のエディタデバッガ4本セットで，エディタでプログラムを組み，アセンブラでコンパイルし，エラーが発生したらデバッガで手直しして……と，こう書くと簡単なのですが，この1つひとつの工程の間にはロード/セーブがあるのです。つまり，エディタのプログラムを起動させ，プログラムを組み，ソースプログラムをセーブし，アセンブラのプログラムを起動させ，ソースプログラムをロードし……。M80に比べるととんでもない遅さでした。こんなことをM80を入手するまで続けていたのですから，今考えると実に恐ろしいことであります。ちなみにM80は決して遅くない。何をするにもイライラするucom82より速い。ようするにディスクへのアクセスタイムが遅いのです。私はいわゆるジャンク屋で，10Mバイトのハードディスクを1個当たり8000円で買ってきて使っています。
松本　剛（20）X1turboZ，PC-1350/1501　神奈川県

●SLANGで気に入っているものに，VAR，CONST，ARRAYなどがある。「宣言しなければ使えない」ことこそプログラマを余計なバグから解放してくれると思うからだ。SLANGはこれからもS-OSの標準構造化言語として使われていくだろう。拡張関数，オプティマイズルーチンの改良など継続的なサポートをお願いしたい。また，「X68000BASIC入門」に表1として挙げられていた「MMLデータ一覧」にシンプルイズベストという意味で花丸をあげたい。X68000のMMLデータが見事にまとめられており，これからX68000でMMLを使おうと思っている人にはたいへんありがたいものだろう。それから，LISPはむずかしいが，「マシン語体操1・2・3番外編」は，LISP講座としては大変よい記事だったと思う。
山口　幸一（22）X1turboII，X68000 ACE，JR-100，PC-1245/55，PC-E200　宮崎県

●Human68kとMS-DOSとの違いは，表面的にはCPUとかアプリケーションの数とかでしょうが，本質的には目指すところが違うと思います。MS-DOSというのはCP/M-86を強く意識して作られていて，まだ16ビットパソコンが出始めのころだったこともあるけれど，パソコンはこうありたい，というポリシーがあまり感じられない。一方，Human68kはX68000とともに，パソコンはこうありたい，というものを考えているなと感じられるのです。MS-DOSマシンやMACのようなマシンに影響されながら，究極のパソコンになろうとしている。そこが素晴しい。
野村　正文（19）MZ-80K，X1D 茨城県

●マシン語を理解できるようになった者としてひと言。「そしてすべてが見えてくる」，ほんとうにそのとおりです。最初はなにがなんだかわからず，なにから始めたらいいのかもわからず困っていました。しかし，ニーモニックで書かれたプログラムを何度も何度も解析し，自分でプログラムをデバッグして，ある日突然，ほんとうに突然，前日までとはまったく違ってすべてが見えてきたのです。マシン語を理解しようとして苦しんでいるとしたら，きっとそれは「すべてが見えてくる」直前なのだと思います。やがて苦しんだかいがあったとわかるでしょう。
佐藤　孝（31）MZ-2500，PC-9801VX41　埼玉県

ごめんなさいのコーナー

6月号　NAMPAシミュレーション
LISP-85の書き換えアドレスがずれていました。
3BA8H　→3BA9H
に変更してください。

6月号　信州
このプログラムをコンパイルするには820行，910行末尾の“｝”を削除し，新たに825行，915行に閉じカッコだけの行を作ってください。

7月号　ポケコンの新しい世界
PC-E200ではPC-1400シリーズのプログラムをテープから読むことはできませんでした。PC-E500では可能です。お詫びのうえ訂正いたします。

バグに関するお問い合わせは
☎03(263)2230(直通)
月～金曜日16：00～18：00

お問い合わせは原則として，本誌のバグ情報のみに限らせていただきます。入力法，操作法などはマニュアルをよくお読みください。
また，よくアドベンチャーゲームの解答を求めるお電話をいただきますが，本誌ではいっさいお答えできません。ご了承ください。

乱,乱,乱数楽しいな…今月の特集は計算だ

▼さぁ夏休みです。どんな計画を立てていますか？　白鳥座や蠍座が夜空の主役になり，夜更かしもいちだんと楽しくなってきましたね。さて，Oh!Xではたっぷり数字と計算を堪能していただこうと，コンピューティングの基礎，数値演算を特集しました。乱数の素顔，πの不思議など，つくづく１＋１が２であるという仮定の偉大さがわかるなぁって思いませんか。

▼今月から登場する連載が３本。まず村田敏幸氏による「Ｚ80マシン語ゲーム工房」。勉強しがいのあるマシン語の記事を，という読者の皆さんの要望にお応えしました。早速感想を聞かせてください。ご意見，ご要望もお待ちしています。

それから１年振りに本誌へ戻ってきた「猫とコンピュータ」。ひょっとしたら人間より賢い(？)あの白猫ホンニャアが，また活躍します。間もなく単行本も出されますのでお楽しみに。

そしてリレー連載によるエッセイ「われら電脳遊戯民」もスタートしました。心底ゲームの好きな彼らのこと，迫力ある内容を期待してください。

▼次に，中森章氏の「X68000BASIC入門」がとうとう最終回を迎えました。X-BASICとX68000の強力な機能をくまなく探索したこの連載は，ユーザーの皆さんにとって強力な味方でしたね。

もうひとつ，浜口勇氏の「実用(？)オブジェクト指向のゲームプログラミング」も今月で終了です。連載中は難解だという声が多数ありましたが，完成にこぎつけたスネークゲームはいかがだったでしょうか。

今後の参考にもいたしますので，ぜひ，ご意見，ご感想などお寄せください。

▼今月の「知能機械概論―お茶目な計算機たち―」と「Between The Lines」は，どちらも筆者が多忙のため，残念ながら休載します。来月は復活できると思いますので，ご容赦のほどを。

▼さぁて，お盆休みをにらんでの締め切り設定を考える頃になると，暑さも本格的になります。かき氷の食べすぎでおなかこわしたりしないようにね。

投稿応募要領

●原稿には，住所・氏名・年齢・職業・連絡先電話番号・機種・使用言語・必要な周辺機器・マイコン歴を明記してください。

●プログラムを投稿される方は，詳しい内容の説明，利用法，できればフローチャート，変数表，メモリマップ（マシン語の場合）に，参考文献を明記し，プログラムをセーブしたテープ（ディスケット）を添えてお送りください。また，プログラムは最低２回はセーブしてください。

●ハードの製作などを投稿される方は，詳しい内容の説明のほかに回路図，部品表，できれば実体配線図も添えてください。編集室で検討の上，製作したハードが必要な場合はご連絡いたします。

●投稿者のモラルとして，他誌との二重投稿，他機種用プログラムを単に移植したものは固くお断りいたします。

あて先

〒102 東京都千代田区九段南2-3-26井関ビル
日本ソフトバンク出版部
Oh！Ｘ「テーマ名」係

SHIFT・BREAK

▶今月はMIDI特集でしたね。結構楽しみにしていた人が多かったんじゃないかなぁ。自分もそのひとりなんだけれども，仲間にいれてもらえなくて残念。ところで，X68000用のMIDI インタフェイスボード早く出ませんかねぇ。一緒にシーケンスソフトなんかついてこないかなぁ。それに加えてMMLで MIDI 出来たりするとMML派の私は最高に嬉しいんですけど。　(善)

▶予備校の夏期講習が始まってますね。去年の私と同じように，予備校生のなかには勉強サボってこれを読んでいる人もいるかもしれないですね。今，君が怠けるのは簡単なことです。でもあとで後悔するのも君自身なんです。受かれとは言わない。また浪人だっていいじゃない。でもベストを尽くしてください，君の受験生時代を悔いないためにも。　(で)

▶いやー，私もやっと自動車免許を取りました。それにしても，免許証の写真撮影って緊張しますね。みんなも学生証の写真撮影なんかで経験あると思うけど，あれって何年も使うもんだから，悲惨な表情に写っちゃうと１年中笑われるんだもんね。よくいるでしょ，殺人犯みたいな顔に写っちゃうやつ。

(クリーミーマミの主題歌を教えてほしいH.K.)

▶最近，みょーに疲れる。年のせいだろうか。まさか！　まだ20歳前だぜ。サークル内の人間関係は難しいし，仕事には追われまくるし，一応勉強もしなくちゃならないし。無理もないな〜。でも遊ぶときになるとやたら元気になる私はいったい何者なんだろう。　(それでも一応幸せなC.W.)

▶「みんなのうた」(サザンのじゃないよ）はかなり凄いということを最近知った。学校に行かなくていいし，仕事をしなくてもいいから「おじいちゃんていいな」というとんでもない歌があったり，山田邦子作詞/歌の「サボテンがこわい」では歌詞もさることながら，バックの粘土アニメが凄い。ロシア民謡風ダンスをするサボテンは必見だね。　(Mu)

▶水族館は奥のビルの10階にあった。昼休みが終わったサラリーマンらとエレベーターへ。10階に着いたのは２人。中ではカップルやおばさんのほかには元気なイルカ，丸い目のアザラシ，腹を見せて泳ぐエイ，餌づけをするギャル，腹の上に子供を乗せて優雅なラッコ。しばし異次元をさまよいしのち，屋上では風切羽を切られて飛べないフラミンゴ。　(K)

▶「伝説巨神イデオン」のビデオが発売になったのでさっそく買ってきた。本放送のときは放映時間がコロコロと変更になって追跡するのが大変だっただけに思い出深い作品だ。当時は物語の進行が遅いという批判が多かった（僕もそう思っていた）が，まとめて観るとちょうどよいテンポだ。これで予告編が収録されていれば完璧だったのに。　(KO)

▶消費税が３％だそうです。例外なく広く薄く課税し，税金分を値引きするよーなことは禁止するそうです。50円のお菓子は51円に，Oh!Xは556円に，少年ジャンプは175円になります。１，５円玉が不足してみんな困るでしょう。政治とはそーゆーものです。ところで自動販売機やゲーセンは103 円になるのでしょうか。どーすんの？　(M)

▶投稿整理にやって来たＨ氏がファイルを探して歩きまわっている。「たぶんＵさんのとこだと思うんですよねー」とＨ氏。でも彼の机の上はダンジョン，いやタワーリングインフェルノだからなぁと，おっかなびっくりのぞきこんだら，あら，ロマンシングザストーンだ。その日，私はＵ氏の机で見つけたのと同じテレマンを６枚とも買ってしまった。　(よ)

▶２回目の締め切りが過ぎた。誰も原稿を持ってこない……。早朝のマシン室で「こんなのREDUCEがあれば一発なんですが」と加藤氏は一所懸命，検算を繰り返す。ゆっくり出社し，ラッシュ前に電車で帰る。１日４食，規則正しい生活。これで帰宅してから食べるのが朝食でさえなかったら，きっと素晴しい生活なんだろうなあ。　(U)

▶もうすっかり夏だというのに，夜行性の私は朝(昼だっけ)起きるのがつらい。だから目覚し時計２個に大音量にしたCDのタイマーセットという強行手段に出たのまではいいのだが，先日のレベッカでは地震かと思って飛び起きたし，大滝詠一では子守歌に聞こえてまた寝てしまった。誰かスッキリと爽快に目が覚める音楽があったら教えてください。　(N)

▶ＸＣの登録ユーザーにはニューバージョンが無料で配布されたそうで，先月号でスッパ抜いた(？)バグも取れている。シャープさんも味なマネをしてくれちゃって。さて，今月の数値演算特集では，難解な数式の嵐に編集室でも目を皿にしたり点にしたり。はっきりいってムズいものはムズいけど，Oh！Xの読者ならきっとがんばって読んでくれるよね。（T）

microOdyssey

いまだにROGUEをやっている。世間はソーサリアンだ，YsIIだ，ファンタジーIIIだと騒がしいが，ストーリータイプのゲームはいまいち手を出す気になれない。毎月，出張校正前には一瞬だけ仕事がなくなるときがある。そういう台風の目のようなときにはROGUEがちょうどいい。イェンダーの護符はまだ遠いが，何年かかってもいつかは究めてみたい。

もともと私がこのゲームを始めようという気になったのは，本誌の前編集長Y氏がこのゲームを評して「ある意味でウィザードリィを超えた」と，珍しく高い評価を下したことによるところが大きい。実際，手を出してみると豊富なアイテムを集めるのは楽しいし，アルファベット1文字のモンスターも非常に個性的に思えてくる。とにかく手軽に手を出せるのがいい。

一時期，市販のコンピュータRPGが経験値稼ぎとアイテム探しのためのモンスター殺戮ゲームと化したことがあった（今でもそうかもしれないが）。そこでその批判として叫ばれたのが「コンピュータRPGにはストーリーがない」ということと「難しくすればいいってものではない」という意見だった。こうして，Ysやドラクエのようなストーリー重視かつ甘口RPGが誕生したのだろう。これはこれで悪くない。むしろ歓迎すべきことであろう。

しかし，これらのゲームで往々にしてわけもわからず行き詰まっている人を見かける。イベントに対しては基本的に一本道なのだ。「単に作者に踊らされているだけじゃないですか」というと「踊らそうという側と踊らされようというプレイヤーがいて，初めてゲームが成立するんじゃないか」。それはもっともだが，RPGってのはそういうもんだっけと思ってしまう。

基本的にROGUEやウィザードリィといったゲームにはストーリーというものが存在しない。だからといって，ドラマがないわけではない。ストーリーがないにもかかわらず，国産RPGのように殺戮ゲームになり下がっていない（そう遊ぶこともできるが）。さて，どこが違うのだろうかと考えてみても，ゲームバランス以外にこれといった大きな違いはないようにも思われる。4M ROMだ，ディスク5枚組だといった物量作戦に頼らなくとも，しっかりとバランスのとれたゲームはオンメモリ，キャラクタ表示のみでも最高に面白いのだ。私はROGUE以上のゲームを知らない。

最近は派手なゲームが多いためか，こういったゲームの本質に関することがおろそかにされているのではないか。はたして，多くの人が現在制作中（あるいは構想中）であろう，「自然会話で進行し，2桁のパラメータを持ち，果てしないマップを持ち，進行状況でシナリオが変化するタイプの究極のRPG」などは必要なのだろうか。なにか方向性を見誤っているのではないか？

ある日のM氏との会話「どうしてROGUEみたいなゲームをS-OS用に作る奴がいないんでしょうね」「単にROGUEを知らないんじゃないですか」うーん，これは残念なことだ。市販ソフトウェアに対する挑戦という意味ではS-OSのようなシステムにこそ，こういった飾りを捨てた本物のゲームが現れてほしいのだが。（U）

1988年9月号8月18日(木)発売

特集　半期に一度のグラフィックバザール

MZ-2500用グラフィックエディタ

C調言語講座PRO-68K　謎の低次元グラフィック

全機種共通システム

WINER用各機種LNPRNTルーチン/超小型エディタ

TED-750/シューティングゲーム　MANKAI

バックナンバー常備店

東京	神保町	三省堂神田本店5F　03(233)3312
	〃	書泉ブックマートB1　03(294)0011
	〃	書泉グランデ5F　03(295)0011
	八重洲	八重洲ブックセンター3F　03(281)1811
	新宿	紀伊国屋書店本店　03(354)0131
	高田馬場	未来堂書店　03(200)9185
	渋谷	大盛堂書店　03(463)0511
	池袋	西武百貨店11Fブックセンター　03(981)0111
	〃	西武百貨店9F コンピュータ・フォーラム　03(981)0111
	町田	久美堂東急ハンズ店　0427(28)2783
神奈川	横浜	有隣堂横浜駅西口店　045(311)6265
	〃	有隣堂ルミネ店　045(453)0811
神奈川	藤沢	有隣堂藤沢店　0466(26)1411
	厚木	有隣堂厚木店　0462(23)4111
	平塚	文教堂四の宮店　0463(54)2880
千葉	柏	新星堂カルチェ5　0471(64)8551
	船橋	西武百貨店10Fブックセンター　0474(25)0111
	〃	芳林堂書店津田沼店　0474(78)3737
	千葉	多田屋千葉セントラルプラザ店　0472(24)1333
埼玉	川越	黒田書店　0492(25)3138
	川口	岩渕書店　0482(52)2190
茨城	水戸	川又書店駅前店　0292(31)0102
大阪	都島区	駸々堂京橋店　06(353)2413
京都	中京区	オーム社書店　075(221)0280
愛知	名古屋	三省堂名古屋店　052(562)0077
	〃	パソコンΣ上前津店　052(251)8334
	刈谷	三洋堂書店刈谷店　0566(24)1134
長野	飯田	平安堂飯田店　0265(24)4545
北海道	室蘭	室蘭工業大学生協　0143(44)6060

定期購読のお知らせ

Oh!Xの定期購読をご希望の方は，最寄りの郵便局にある払込用紙に，

口座番号　東京1-29307

加入者名　株式会社日本ソフトバンク

とご記入のうえ，年間購読料6,500円を添えてお申し込みください。その際，裏面の通信欄に「○年○月号よりOh!X定期購読希望」と忘れずに明記してください。なお，すでに定期購読をご利用いただいている方には，購読期限終了と同時にご通知申し上げますので，同封の払込用紙をご利用ください。

海外送付ご希望の方へ

本誌の海外発送代理店，日本IPS(株)にお申し込みください。なお，購読料金は郵送方法，地域によって異なりますので，下記宛必ずお問い合わせください。

日本IPS株式会社
〒101 東京都千代田区飯田橋3-11-6
☎03(238)0700

Oh!X 8月号

■1988年8月1日発行　定価540円　■発行人　孫　正義　■編集人　笹口幸男

■発行元　(株)日本ソフトバンク

■出版事業部　〒102 東京都千代田区九段南2-3-26 井関ビル　☎03(261)4095　FAX 03(262)8397

編集室☎03(239)4156
出版営業☎03(261)4095
広告営業☎03(297)0181

■本　社　〒102 東京都千代田区九段南2-3-14　靖国九段南ビル　☎03(263)3690㈹
TELEX 東京　232-4614JSBTYJ　FAX 03(263)3660

■西日本営業部　〒541 大阪府大阪市東区南本町2-6　明治生命堺筋本町ビル10F
☎06(264)1471㈹　FAX 06(264)1481

■印　刷　凸版印刷株式会社

本物かどうかが
超多機能の条件。

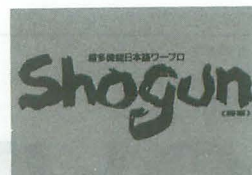
定価¥34,800

X1シリーズ用ワープロNo.1
(株)日本ソフトバンク刊「Oh！新作売れ筋」(～Vol.181)の全国売り上げランキング調査による)

16ビット用最新、自動/一括/連文節変換システムKatana(刀)の完全移植。143万種にも及ぶ多彩な文字表現。※1本格的データベース、表計算機能搭載。16ビットワープロソフト、データベースソフトなどMS-DOS上で動くソフトとのデータ互換。※2その他すべての機能が16ビット用に開発されたパーツ群により構成。フルスペックでなおかつ超高速。

※1. 文字サイズ・文字種・文字の位置・網かけ・下線・カラー設定の組みあわせによる計算。※2. MS-DOSとのデータ交換は2HD版のみ。※MS-DOSはマイクロソフト社の登録商標です

- Katana(刀)が自動・一括・連文節変換実現。
- カード型データベース機能、表計算機能搭載。
- 他の追従を許さぬ文字表現力。
- 多様な用紙への印刷が可能。

超多機能日本語ワープロ Shogun

SHARP X1 turbo III / Z 用2HD版
SHARP X1 turbo シリーズ対応2D版
※本商品はX1ではお使いいただけません。あらかじめご了承ください

人を大切にするテクノロジー
株式会社 サムシンググッド
〒160 東京都新宿区大久保2.5.20シティプラザ新宿
TEL.03(232)0801(代表)

《広告の半ページ》　相変わらずウソくさいけど本当に売ってます。

月刊 電脳倶楽部　8月号(Vol.3)　7月18日発売

2HDディスクに入ったX68000のための雑誌だっ！

納涼ソフトで夏が来るっ！

花火ソフト：スターマインから線香花火を経由してヘビ花火まで

ホラー，オカルト，スプラッタソフト！

パソコンのはらわた

PDS

- ATTRIB.Xをパワーアップ：ATTRIBED.X
- EJECT.X, FF.X, PCMSTOP.X, OFF.X, TVCTRL.Xなど，ピリリと小技をキメる
- タテ・ヨコパズルジェネレータ

BASIC外部関数

- キーバッファしてしまうkey0( )
- EJECT( )など，もう一度ピリリと小技をキメる
- MSXマウスをつないでしまったりする

RATドライバの製作－その1

- 周辺機器争奪バトルロイヤル参加者募集
- その他，色々思いつくままにやってしまうのだ

さらには謎のプレゼントも炸裂

もちろん，これ以外にてんこもりっ！

なお，内容は一部変更されることがあります。御了承ください。

編集長祝一平からの御挨拶「これから暑い夏がやって来ますが私のせーではありません」

満開製作所　電脳倶楽部 編集部

〒171 東京都豊島区要町1－3－24 三浦ビル3F
TEL.(03)554－9282(いたずら電話はしないでね)

販売方法は通信販売のみです。お申し込みの方法は左記の住所へ現金書留で

◆ **定期購読 6ヵ月分 6,000円**(郵送料サービス)

◆ 何月号(Vol.何？)からの購読かを明記してください。創刊号に逆上っての購読もお受けしております。

◆ 郵便振替を御利用の場合は口座番号「東京 5-362847 満開製作所」でお願いいたします。
(振替を御利用の場合は発送までに10日以上かかります)

シャープコーナーさらに充実！
夏休みお楽しみセール7/20～8/20
●お買上げの際学生証をご提示の方、ゲームソフト他サプライ品20%OFF！
●期間中1万円以上お買上げの方にステキな景品プレゼント！

営業時間
AM10:00～PM7:00
（日曜・祭日はPM6:00まで）
年中無休

これからは16ビット！X68000

START → お好きな組合せでどうぞ。

本体

スタンダードモデル
X68000
・CZ-601C（E・B）¥319,800

プロフェッショナルタイプ
X68000 ACE HD
・CZ-611C-GY … ¥399,800
新製品・20Mハードディスク内蔵!!

ディスプレイ

●CZ-601D-GY（BK）‥ ¥119,800
ピッチ0.39・アナログ対応

CZ-611D-GY（BK）‥ ¥145,000
ピッチ0.31・アナログ対応

●CU-15M1（E・B）‥‥‥¥ 99,800
ピッチ0.39・アナログ/デジタルモニター

周辺機器・ボード

●CZ-8PK7‥‥‥‥‥‥‥¥122,000
80桁ドットインパクトプリンター
●CZ-8PK8‥‥‥‥‥‥ ¥152,000
136桁ドットインパクトプリンター
●CZ-8PC2‥‥‥‥‥‥ ¥ 69,800
80桁熱転写プリンター
●CZ-6BE1(A)‥‥‥‥ ¥ 35,000
1MB増設RAM（CZ-601C用）
★その他いろいろあります。お電話で！

ソフト PART 1

●日本語ワープロEW‥‥¥38,000
フロントプロセッサE1搭載ワープロソフト
●WINDEX PRO-68K・¥28,000
コンパイラと来たらエディタです。
●Kamikaze‥‥‥‥‥‥‥ ¥68,000
忘れちゃいけないビジネスソフト
●Z'S STAFF PRO-68K‥¥58,000
プロフェッショナルグラフィックツール

ソフト PART 2

●XLink 68‥‥‥‥‥‥‥‥¥19,800
時代はパソコン通信だ！
●ミュージックPRO-68K‥ ¥18,800
●サウンドPRO-68K‥‥¥15,800
ミュージック関係ならこの2本！
●サンプリングPRO-68K‥¥17,800
PCMをフル活用するならこれ！
●C-TRACE68000‥‥‥¥68,000
本格的レイトレーシングツール

GOAL
さあ、ご注文、お問合せは今スグお電話で！お支払いは超低金利のクレジットもご利用できます。お気軽にご連絡ください。
☎03-486-6541
ソフトやハードの内容や発売日等のおたずねにも親切にお答えします。

組合せのほんの一例
名づけて‥必殺！ゲーマーズセット

●CZ-601C（E・B）本体＋キーボード‥‥¥319,800
●CZ-601D（E・B）ディスプレイテレビ¥119,800
●CZ-6ST1（E・B）チルトスタンド‥‥¥ 5,800
●スペースハリアー‥‥‥‥‥‥‥‥‥ ¥ 6,800
●源平討魔伝‥‥‥‥‥‥‥‥‥‥‥‥ ¥ 7,800
●XE-1PRO（ジョイスティック）‥‥‥ ¥ 9,500
■定価合計‥‥‥‥‥‥‥‥‥‥‥‥‥¥528,700

均等払い	ボーナス
¥16,850×18回	¥30,000× 3回
¥12,670×24回	¥25,000× 4回
¥ 8,490×36回	¥20,000× 6回

●ソフトも周辺機器も紹介しきれないぐらい豊富です。くわしくはお電話で！

それでもやっぱりX1だ！

X1 turbo Z II

●CZ-881C-BK本体＋キーボード‥¥179,800
●CZ-880D-BKディスプレイテレビ‥¥109,800
●CZ-6ST1B チルトスタンド‥‥‥¥ 5,800
●AN-160SPアンプ内蔵スピーカー・¥ 59,800
●ブランクディスケット‥‥‥‥ ¥ 4,500
■定価合計 ‥‥‥‥‥ ~~¥359,700~~

X1 twin

●CZ-830C-BK本体＋キーボード・¥99,800
●CZ-830D-BKディスプレイテレビ‥‥¥98,000
●CZ-6ST1B チルトスタンド‥‥‥‥ ¥ 5,800
●上海（ゲームソフト）‥‥‥‥‥ ¥ 4,500
●ブランクディスケット‥‥‥‥‥‥ ¥ 4,500
■定価合計 ‥‥‥‥‥‥ ~~¥212,600~~

クリエイト特典
●全商品完全保証書付（メーカー保証）
●全国無料配達（一部離島の方は有料になります）
●配達日の指定OK（日曜・祭日にかかわらずお客様のご都合にあわせて配達します）
●どんな商品の組合せも自由自在（ご予算、用途に応じ自由自在にシステムアップできます）
●中古パソコン高額下取（今お使いのパソコンをわずかな差額でグレードアップ）
●お支払い方法自由（低金利の均等払、ボーナス一括払もご利用下さい）

大特価周辺機器（各ケーブル付き）

品名	定価	機能説明
ITH-320S	~~¥125,000~~	20Mハードディスク 平均シークタイム 28ms以下
ITH-520N	~~¥ 99,800~~	20Mハードディスク 平均シークタイム 65ms以下
ITH-540S	~~¥168,000~~	40Mハードディスク 平均シークタイム 38ms以下
VP-800	~~¥122,000~~	80桁シリアルプリンタ

当店はX68000の認定店です。どんなことでも安心してご相談ください。
★X68000をお買上げのお客様にもれなくテレホンカードとゲームソフト（アルカノイド）をプレゼント中！

総合お問合せ先☎03-486-6541㈹

●横浜店
駅から歩いて2分！

パソコン専門ショップ
ソフトクリエイト 渋谷/横浜

●渋谷店☎03-486-6541（代）
〒150:東京都渋谷区渋谷1-12-7 三和渋谷ビル
振込銀行:協和銀行 渋谷支店㊟No.239313

●横浜店☎045-314-4777（代）
〒221:横浜市神奈川区鶴屋町2-12-8 第1建設ビル
振込銀行:三和銀行 横浜駅前支店㊟No.310852

秋葉原価格でローンができます
電気の街秋葉原で
24年
の信用!!

AVCフタバ

AUDIO VIDEO COMPUTER

☎03(253)7661

至新宿
電波ビル
リビナ
至神田
中央通り
秋葉原駅
AVCフタバ電機通信販売部
外堀通り
AVCフタバ電機本社
日通ビル
至上野
ヤマギワ

AVCフタバ電機
〒101 東京都千代田区外神田2-9-8
神田ユニオンビル ☎03-253-7661(代)

今すぐ もよりの電話から	仙 台 022-264-3704	名古屋 052-452-3271	広 島 082-295-6873
札 幌 011-611-5104	新 潟 0252-75-4175	大 阪 06-311-3931	福 岡 092-481-2494

X68000 ACE-HD

ドットピッチ0.31mmのモニターをセット20MHD搭載の超高級セット。

CZ-611C……¥399,800
CZ-611D……¥145,000
合計………¥544,800

特価 ¥4?4,000

お支払例 ¥40,145×12回 ¥21,338×24回 ¥14,949×36回 ¥11,754×48回

X68000 ACE-HD

強力な日本語処理と実装密度を追求して信頼性、更に向上。

CZ-611C……¥399,800
CZ-601……¥119,800
合計………¥519,600

特価 ¥4?4,000

お支払例 ¥38,295×12回 ¥20,355×24回 ¥14,260×36回 ¥11,213×48回

X68000 ACE-HD

更に夢を拡大、20MB HDの搭載。最大に能力を引出す3モードのディスプレイ。

CZ-611C……¥399,800
CU-15M1……¥ 99,800
合計………¥499,600

特価 ¥4?0,000

お支払例 ¥37,000×12回 ¥19,667×24回 ¥13,778×36回 ¥10,833×48回

X68000 ACE

CZ-600の後継 直接アクセスできるメモリーが16MBもある優れもの。

CZ-601C……¥319,800
CZ-601D……¥119,800
合計………¥439,600

特価 ¥3?3,000

お支払例 ¥32,653×12回 ¥17,356×24回 ¥12,159×36回 ¥ 9,560×48回

X1turboZII

X1turboZの本格派セット。TV付2モードオートスキャンディスプレイ。

CZ-881C……¥179,800
CZ-881D……¥109,800
合計………¥289,600

特価 ¥2?3,000

お支払例 ¥20,628×12回 ¥10,964×24回 ¥ 7,681×36回 ¥ 6,040×48回

X1turboZII

NEW-Z BASICの搭載でAV機能をサポート、充分に楽しめるぞ。

CZ-881C……¥179,800
CU-14BD……¥ 64,800
合計………¥244,600

特価 ¥19?,000

お支払例 ¥17,760×12回 ¥ 9,440×24回 ¥ 6,613×36回 ¥ 5,200×48回

X68000

20MB大容量メモリ内蔵可能、しかも強力な日本語処理、夢を超えたパソコンだ!

CZ-600C……¥369,000
CZ-600D……¥129,800
合計………¥498,800

特価 ¥3?8,000

お支払例 ¥31,265×12回 ¥16,618×24回 ¥11,642×36回 ¥ 9,154×48回

X68000 ACE

CZ-600の後継 直接アクセスできるメモリーが16MBもある優れもの。

CZ-601C……¥319,800
CU-15M1……¥ 99,800
合計………¥419,600

特価 ¥3?5,000

お支払例 ¥30,988×12回 ¥16,471×24回 ¥11,539×36回 ¥ 9,073×48回

mZ-2861

高級日本語 ワープロ「書院28」搭載、MS-DOSと融合しスピーディな実務。

MZ-2861……¥328,000
MZ-1D26……¥ 89,800
合計………¥417,800

特価 ¥3?3,000

お支払例 ¥30,803×12回 ¥16,373×24回 ¥11,470×36回 ¥ 9,019×48回

mZ-2531

今よりもなおハイグレードに、とお考えの方に…更に可能性を拡大する。

MZ-2531……¥199,800
MZ-1D22……¥108,000
合計………¥307,800

MZ-2531▲

特価 ¥2?8,000

お支払例 ¥22,015×12回 ¥11,702×24回 ¥ 8,198×36回 ¥ 6,446×48回

mZ-2520

MZをこれから使う方、少々ぜいたくですが是非お勧めしたいセットです。

MZ-2520……¥159,800
MZ-1D26……¥ 89,800
合計………¥249,600

特価 ¥2?0,000

お支払例 ¥18,500×12回 ¥ 9,833×24回 ¥ 6,889×36回 ¥ 5,417×48回

X1turboZ

NEW-Z BASICは後で買えばいい。ハイグレードモニタをセットして驚異の価格。

CZ-880C……¥218,000
CZ-880D……¥109,800
合計………¥327,800

特価 ¥1?8,000

お支払例 ¥16,465×12回 ¥ 8,752×24回 ¥ 6,131×36回 ¥ 4,821×48回

X1Gmodel30

X1Gの本格派セットFDD2基内蔵、専用カラーモニタはTVにも使用可能。

CZ-822C……¥118,000
CZ-820D……¥ 79,000
合計………¥197,000

特価 ¥??9,800

お支払例 ¥ 9,232×12回 ¥ 6,348×18回 ¥ 4,906×24回 ¥ 3,438×36回

X1Gmodel10

FDDも欲しい、レコーダも使いたい、予算が10万以下ならこのセットをお勧めします。

CZ- 820C…¥ 69,800
CZ- 820D…¥ 79,000
CZ- 503F…¥ 49,800
合計………¥198,600

特価 ¥94,800

お支払例 ¥ 8,769×12回 ¥ 6,030×18回 ¥ 4,661×24回 ¥ 3,265×36回

X1Fmodel10

とりあえず BASIC を覚えてディスクソフトで遊ぶのは後まわしそんな君へのセット。

CZ-811C……¥ 89,800
CU-14G……¥ 49,800
合計………¥139,600

特価 ¥59,800

お支払例 ¥10,664×6回 ¥ 6,578×10回 ¥ 5,532×12回 ¥ 3,804×18回

X1twin

HEシステムを搭載、最上級ゲーム機とパソコンが合体。

CZ-830C……¥ 99,800
CZ-830D……¥ 98,000
合計………¥197,800

特価 ¥1?8,000

お支払例 ¥14,615×12回 ¥ 7,768×24回 ¥ 5,442×36回 ¥ 4,279×48回

●分割回数は3回〜48回まで自由に選べます。

●セットの組合せは自由!広告に出ていない他の機種はお問合せ下さい。

型番	品名	標準価格	販売価格	お支払例
CU-14GE	ディスプレイ	¥ 49,800	¥ 29,800	¥ 3,278×10回
CU-14BD	ディスプレイ	¥ 64,800	¥ ?7,000	¥ 3,541×15回
CU-14A4	ディスプレイ	¥ 89,800	¥ ?3,000	¥ 3,371×18回
CU-14AD	ディスプレイ	¥ 84,800	¥ ?8,000	¥ 3,689×18回
CU-15M1	ディスプレイ	¥ 99,800	¥ ?7,000	¥ 3,786×24回
CZ-820D	ディスプレイ	¥ 79,800	¥ ?4,800	¥ 3,375×15回
CZ-880D	ディスプレイ	¥109,800	¥ 8?,000	¥ 4,081×24回
CZ-601D	ディスプレイ	¥119,800	¥ 9?,000	¥ 3,203×36回
CZ-611D	ディスプレイ	¥145,000	¥1?3,000	¥ 3,892×36回
CZ-520F	FDD(2HD/2DD)	¥118,000	¥ 9?,000	¥ 3,169×36回
CZ-502F	FDD(2DD)	¥ 99,800	¥ 7?,000	¥ 3,172×30回
CZ-503F	FDD(2D)	¥ 49,800	¥ 3?,800	¥ 3,219×12回
CZ-6BE1	1MB 増設RAMボード	¥ 35,000	¥ 2?,000	¥ 3,080×10回
CZ-6BE2	2MB 増設RAMボード	¥ 79,800	¥ 6?,000	¥ 3,944×18回
CZ-6BE4	4MB 増設RAMボード	¥138,000	¥1?8,000	¥ 3,720×36回

型番	品名	標準価格	販売価格	お支払例
CZ-620H	HDD	¥178,000	激安	? × ?
CZ-8PC2	プリンタ(80桁)	¥ 69,800	¥ 5?,000	¥ 4,995×12回
CZ-8PK5	プリンタ(80桁)	¥129,000	¥1?0,000	¥ 3,444×36回
CZ-8PK7	プリンタ(80桁)	¥122,000	¥ 9?,000	¥ 3,238×36回
CZ-8PK6	プリンタ(136桁)	¥159,000	¥1?3,000	¥ 3,331×48回
CZ-8PK8	プリンタ(136桁)	¥152,000	¥1?7,000	¥ 3,169×48回
CZ-8PK9	プリンタ(80桁)	¥ 89,800	¥ ?0,000	¥ 3,442×24回
CZ-8NS1	カラーイメージスキャナ	¥188,000	¥1?0,000	¥ 4,063×48回
CZ-6VT1	カラーイメージユニット	¥ 69,800	¥ 5?,000	¥ 3,562×18回
CZ-6PV1	カラービデオプリンタ	¥198,000	¥1?4,000	¥ 4,171×48回
AN-160SP	アンプ内蔵スピーカー	¥ 59,800	¥ ?8,000	¥ 3,053×18回
CZ-8BS1	FM音源ボード	¥ 23,800	¥ 19,800	現金一括払
CZ-6BN1	スキャナ用パラレルボード	¥ 29,800	¥ 24,000	現金一括払
CZ-6BU1	ユニバーサル10ボード	¥ 39,800	¥ 3?,000	¥ 3,520×10回
CZ-6BG1	GP-IBボード	¥ 59,800	¥ 4?,000	¥ 3,053×18回

型番	品名	標準価格	販売価格	お支払例
CZ-6BF1	増設RS232Cボード	¥ 49,800	¥ 4?,000	¥ 3,013×15回
CZ-6BP1	数値プロセッサボード	¥ 79,800	¥ 6?,000	¥ 3,147×24回
CZ-6EB1	I/Oボックス	¥ 88,000	¥ 6?,000	¥ 3,343×24回
CZ-8TM1	モデム	¥ 29,800	¥ 2?,000	現金一括払
CZ-8TM2	モデム	¥ 49,800	¥ ?9,000	¥ 3,608×12回
CZ-213MS	MUSIC PRO-68K	¥ 18,800	¥ 15,800	現金一括払
CZ-214MS	SOUND PRO-68K	¥ 15,800	¥ 13,800	現金一括払
CZ-212BS	ビジネス PRO-68K	¥ 68,000	¥ 5?,000	¥ 3,435×18回
CZ-211LS	Cコンパイラ PRO-68K	¥ 39,800	¥ 3?,000	¥ 3,520×10回
CZ-141SF	NEW-Z BASIC	¥ 18,800	¥ 15,800	現金一括払
CZ-137SF	turbo Z's STAFF	¥ 19,800	¥ 16,800	現金一括払
CZ-133SF	モデムターミナルソフト	¥ 25,800	¥ 2?,000	現金一括払
	Z'STAFF PRO-68K	¥ 58,000	¥ 47,000	¥ 3,541×15回
	kamikaze	¥ 68,000	¥ 5?,000	¥ 3,499×18回
	Shogun	¥ 34,800	¥ 30,000	現金一括払

頭金なし 手軽な電話クレジット。

カレッジクレジット 保証人なし。但し満20才以上の学生の方。

納 期 通常の場合、当社に申込書が到着後1週間以内。特に人気のある商品で品薄の場合、少々納期が遅れる場合もありますので御了承下さい。

製品先取り お支払いは約1〜2カ月後から。

低金利クレジット 1回の支払は2,700円以上で3〜48回。ボーナス併用も可。

完全保証 すべてメーカー保証書付 アフターケア万全。

全国代引 お届けした者に、代金をお支払いいただく方法です。(但し、手数料1,000円)

18才未満の方 ご両親が代理購入者としてお申し込み下さい。

AM10時からPM8時まで受付 日曜・祝日も営業

またまた秋葉原でおなじみの

7/15〜8/20

●お近くの方はお
●本体単品で特
●ビジネスソフト定

さらに金利が安くなった!! 超低金利クレジット

回数	金利
▶12回	4.5%
▶24回	9.5%
▶36回	13.0%
▶48回	17.0%
▶60回	22.0%

X68000ACE HD (送料¥2,000)

Ⓐセット:
CZ-611C+CZ-611D+M-2HD(10枚)
……定価¥544,800➡P&A超特価(価格はお電話下さい。)

12回	37,000	24回	19,300	36回	13,300	48回	10,300	60回	8,600

Ⓑセット:
CZ-611C+CZ-601D+M-2ND(10枚)
……定価¥519,600➡P&A超特価(価格はお電話下さい。)

12回	34,800	24回	18,200	36回	12,500	48回	9,700	60回	8,100

※X-68000セットでお買い上げの方に源平討魔伝¥7,800をプレゼント致します。
※チルトスタンド(CX6ST¥5,800)必要な方は¥5,000加算して下さい。

X68000ACE (送料¥2,000)

Ⓐセット:
CZ-601C+CZ-611D+M-2HD(10枚)
……定価¥464,800➡P&A超特価(価格はお電話下さい。)

12回	31,300	24回	16,400	36回	11,300	48回	8,700	60回	7,300

Ⓑセット:
CZ-601C+CZ-601D+M-2HD(10枚)
……定価¥439,600➡P&A超特価(価格はお電話下さい。)

12回	29,600	24回	15,500	36回	10,600	48回	8,200	60回	6,900

※チルトスタンド(CZ-6STI ¥5,800)必要な方は¥5,000加算して下さい。
※X-68000セットでお買い上げの方に源平討魔伝¥7,800をプレゼント致します。

X-1ターボZ/ZII (送料¥2,000)

※チルトスタンド(CZ-6STI¥5,800)必要な方は¥5,000加算して下さい。

Ⓐセット:
X-1ターボZ(CZ-880C+CZ-880D)+M-2HD(10枚)+ジョイカード+ゲームソフト3種
……定価¥327,800➡超特価**¥180,000**

12回	15,600	24回	8,200	36回	5,600	48回	4,300	60回	3,600

●NEW Z-BASIC(CZ-141SF¥18,800)必要な方は、¥15,000加算して下さい。

Ⓑセット:
X-1ターボZII(CZ-881C+CZ-880D)+M-2HD(10枚)+ジョイカード+ゲームソフト3種
定価¥289,600➡P&A超特価(価格はTel下さい。)

12回	18,200	24回	9,500	36回	6,500	48回	5,100	60回	4,200

ソフトコーナー (送料¥1,000)

X-68000用

Ⓐ CZ-213MS(MUSIC PRO-68K)……定価¥18,800➡特価**¥16,000**
Ⓑ CZ-214MS(SOUND PRO-68K)……定価¥15,800➡特価**¥13,500**
Ⓒ CZ-212BS(BUSINESS PRO-68K)……定価¥68,000➡特価**¥57,000**
Ⓓ CZ-211LS(C compiler PRO-68K)……定価¥39,800➡特価**¥31,500**
Ⓔ Z's STAFF PRO-68K(シャフト)……定価¥58,000➡特価**¥46,000**
Ⓕ Kamikaze(神風)(サムシンググッド)……定価¥68,000➡特価**¥49,000**
Ⓖ ビジネスAD68K(マッシュシステム)……定価¥98,000➡特価**¥78,500**
Ⓗ 弥生(日本マイコン販売)……定価¥80,000➡特価**¥64,000**
Ⓘ CP/M-68K(ニューウェイブ)……定価¥110,000➡特価**¥88,000**

X-1シリーズ

Ⓙ SHOGUN(サムシンググッド)……定価¥34,800➡特価**¥25,000**
Ⓚ SAMURAI(サムシンググッド)……定価¥19,800➡特価**¥15,200**

P&Aがズバリ超特価セールでご奉仕!!

立寄り下さい。専門係員が説明いたします。
価で受付します。詳しくは電話にてお問合せ下さい。
価の20%引きOK! TELください。

全国通販

★頭金なし! ★即日発送

超低金利クレジットOK!! 1回〜60回払いまでOK!!

X-1twin (送料¥2,000)

Ⓐセット:
X-1twin(CZ-830+RFコンバーター(AN-58C)+M-2D(10枚)+ジョイカード+ゲーム3種
……定価¥102,780➡超特価**¥77,000**

12回	6,700	24回	3,500

Ⓑセット:
X-1twin(CZ-830C+CZ-830D)+M-2D(10枚)+ジョイカード+ゲーム3種
定価¥197,800➡超特価**¥142,000**

12回	12,300	24回	6,400	36回	4,400	48回	3,400

X-1Gモデル30 (送料¥2,000)

Ⓐセット:
X-IGモデル30(CZ-822+RFコンバーター(AN-58C)+M-2D(10枚)+ジョイカード+ゲーム3種…………
…定価¥120,980➡超特価
¥62,000

12回	5,300	20回	3,300

Ⓑセット:
X-IGモデル30(CZ-822C+CZ-820D)+M-2D(10枚)+ジョイカード+ゲーム3種…………
定価¥197,800➡超特価**¥98,000**

12回	8,500	24回	4,400	36回	3,000

プリンターセット ※全セットにケーブル、用紙付 (送料¥1,000)

Ⓔ CZ-8PK6 限定品
定価159,000
↓
特価**¥89,800**
(用紙1000枚付 送料無料)

Ⓕ CZ-8PK5 限定品
定価129,000
↓
特価**¥69,800**
(用紙1000枚付 送料無料)

Ⓐセット:CZ-8PC2…………定価¥69,800➡超特価**¥55,000**

12回	4,600	18回	3,200

Ⓑセット:CZ-8PK7…………定価¥122,000➡P&A超特価

12回	8,100	24回	4,200	30回	3,500

Ⓒセット:CZ-8PK8…………定価¥152,000➡P&A超特価

12回	10,100	24回	5,300	36回	3,600

Ⓓセット:CZ-8PK9…………定価¥89,800➡P&A超特価

12回	6,000	24回	3,100

カラービデオプリンター (送料¥1,000)

Ⓐセット:CZ-6PVI…………定価¥198,000➡超特価**¥155,000**

12回	13,400	24回	7,000	36回	4,800	48回	3,700

カラーイメージスキャナ (送料¥1,000)

Ⓐセット:CZ-8NSI…………定価¥188,000➡超特価**¥145,000**

12回	12,600	24回	6,600	36回	4,500	48回	3,500

周辺機器コーナー(送料¥1,000)●その他の周辺機器はお電話下さい。

- ⒶCZ-8BSI(FM音源ボード)…………定価¥23,800➡特価**¥19,000**
- ⒷCZ-8RLI(データレコーダ)…………定価¥24,800➡特価**¥20,000**
- ⒸCZ-8AV2(カラーイメージボードⅡ)…………定価¥39,800➡特価**¥31,000**
- ⒹCZ-8BRI(立体映像セット)…………定価¥29,800➡特価**¥23,000**
- ⒺCZ-8DT2(パーソナルテロッパ)…………定価¥44,800➡特価**¥35,000**
- ⒻCZ-6VTI(カラーイメージユニット)…………定価¥69,800➡特価**¥55,000**
- ⒼCZ-6EBI(拡張I/Oボックス)…………定価¥88,000➡特価**¥69,000**
- ⒽAN-160SP(アンプ内蔵スピーカーシステム)…定価¥59,800➡特価**¥47,000**

通信モデムコーナー (送料¥1,000)

- ⒶPV-A1200MKⅡ(アイワ)…………定価¥26,800➡特価**¥21,000**
- ⒷPV-A2400MNP(アイワ)…………定価¥59,800➡特価**¥44,000**
- ⒸMD-1200AⅡ(オムロン)…………定価¥24,800➡特価**¥19,000**
- ⒹMD-2400B(オムロン)…………定価¥49,800➡特価**¥37,500**
- ⒺSR-120S(エプソン)…………定価¥29,800➡特価**¥23,000**
- ⒻSR-240AT(エプソン)…………定価¥59,800➡特価**¥47,000**

P&A 特選パソコンラック (送料無料)

Ⓐ3段 875(H)×580(D)×610(W) **¥8,500**
Ⓑ4段 1245(H)×600(D)×614(W) **¥13,500**
Ⓒ5段 1280(H)×600(D)×620(W) **¥16,500**

通信販売お申し込みのご案内

〔現金一括でお申し込みの方〕
●商品名およびお客様の住所・氏名・電話番号をご記入の上、代金を当社まで、現金書留でお送りください。(プリンター・フロッピーの場合、本体使用機種名を明記のこと)
〔銀行振込でお申し込みの方〕
●銀行振込ご希望の方は必ずお振込みの前にお電話にてお客様のご住所・お名前・商品名等をお知らせください。
(電信扱いでお振込み下さい。)

〔振込先〕住友銀行 新小岩支店
当No.263914 ㈱ピー・アンド・エー

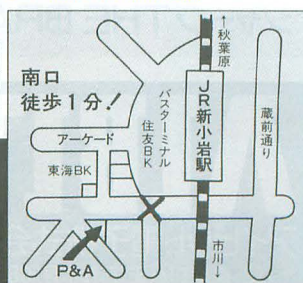
〔クレジットでお申し込みの方〕
●電話にてお申し込みください。クレジット申し込み用紙をお送りいたしますので、ご記入の上、当社までお送りください。
●現金特別価格でクレジットが利用できます。残金のみに金利がかかります。
●1回〜60回払いまで出来ます。但し、1回のお支払い額は3,000円以上。

アフターサービス万全
全商品保証付。専門の担当者がお客様の立場で対応します。
初期不良、輸送トラブルetc.
万が一初期不良、輸送トラブルが発生しました際には、即交換させていただきます。

●定休日/毎週水曜日=第3水曜・木曜は連休とさせていただきます(祭日の場合は翌日になります)

超低金利クレジット率

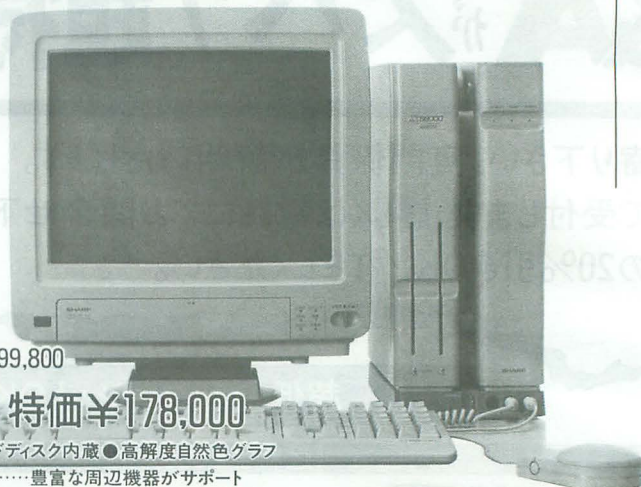
回数	1	3	6	10	12	15	18	24	36	48	60
利率(%)	1.5	2.0	3.0	4.5	4.5	7.5	9.0	9.5	13	17	22

南口 徒歩1分!
秋葉原↑ JR新小岩駅 市川↓ 蔵前通り バスターミナル 住友BK アーケード 東海BK P&A

●マイコン ●ビデオ ●ビデオテープ

株式会社ピー・アンド・エー
〒124 東京都葛飾区新小岩2丁目7番地1号
☎03-651-0148(代) FAX 03-651-0141
●営業時間 AM11:00〜PM9:00
日・祭日も受付けます。
(但しPM8:00迄)

価値あるアイビット、パワーアップ下取りセール！

いま御使用のパソコンを高価下取りの上、MZ-6500モデル50、シャープX68000ACE HD、またはパソコンテレビX1turboZをアイビットならではのサービス大特価でお届けします。

20Mバイトハードディスク搭載、クリエイティブワークステーションX68000が、いま熱い

シャープX68000ACE HD

本体＋キーボード(CZ-611C-GY) 標準価格￥399,800

CZ-600Cを下取りした場合 特価￥178,000

●68000搭載●最大12バイトの大容量メモリ●20Mバイトハードディスク内蔵●高解度自然色グラフィックス●フレンドリーOS Human68K搭載等、先進機能満載。……豊富な周辺機器がサポート

セット大特価！ズバリ￥340,000！

●本体＋キーボード(CZ-600E) 標準価格~~￥369,000~~
●15型カラーディスプレイテレビ(CZ-600DE) 標準価格~~￥129,800~~
●チルトスタンド(CZ-6ST1E) 標準価格~~￥5,800~~

高度なパフォーマンスを秘めて、新登場！

12MHzの高速80286CPU、高速グラフィックLSI搭載。
AI辞書による高度な日本語処理、MS-DOS V3.1。

MZ-6500 モデル50

●16ビットパーソナルコンピュータMZ-6551
(1.2MB FD2基搭載) 標準価格￥430,000
●16ビットパーソナルコンピュータMZ-6556
(1.2MB FD2基、20MB HD1基搭載) 標準価格￥650,000

☆下取り価格、及び特価につきましては、電話でお問い合わせください。

リアルな映像と音が創造力を刺激する。

パソコンテレビX1turboZ

本体＋キーボード(CZ-880C)
標準価格￥218,000

下取り機種問わず/サービス大特価￥100,000

●アナログカラーイメージボード内蔵●4,096色対応ニューテロッパ機能●8重和音ステレオFM音源搭載●マウス標準装備●JIS第1/第2水準漢字ROM実装●システム、ユーザー辞書装備●1Mバイト5インチフロッピー2基搭載。

お買い上げの方にプレゼント

●モデムユニット(シャープCZ-8TM1)

新製品！ ワープロの枠を越え、スケジュール管理機能、アドレス管理機能等を搭載。

ノートワープロ
シャープTHE BRAIN(WV-500) 標準小売価格 ￥138,000⇒特価￥125,000

アイビット電子株式会社
〒192 東京都八王子市北野町560-5
☎0426-45-3001～3
FAX.0426-44-6002
●営業時間：10:00～19:00
●電話受付：20:00迄可
●定 休 日：日曜日(祭日営業)

信用をモットーに、よりよい品をより安く、迅速にお届けします。

全国通信販売 北海道から沖縄まで

★送料はご注文の際にお問い合わせ下さい。
★掲載の商品は、すべて新品、保証書付きです。
★掲載の商品は充分用意してありますが、ご注文の際は、在庫の確認の上、現金書留または、銀行振込でお申し込み下さい。全商品クレジットでも扱っております。
★お申し込みの際は必ず電話番号を明記して下さい。
★商品、品切れの節はご容赦下さい。

富士銀行八王子支店 (普)1752505

■富士通FM周辺機器（約150機種）ご希望の方に特価リスト差し上げます。（切手70円を添えてお申し込み下さい。）

他にもカラーモニター各種在庫あります。

圧倒のX1・MZ品揃えの豊かさ、安さが自慢。

本誌発売時には、下記価格表より、さらにお求めやすい価格に変更されている場合があります。

本体

- ●シャープCZ-820CX1G………￥69,800⇨￥19,800
- ●シャープ600C+600D+6ST1セット￥504,600⇨￥340,000
- ●シャープCZ-601CX68000ACE‥￥319,800⇨超特価☎
- ●シャープCZ-611CX68000ACE HD￥399,800⇨超特価☎
- ●シャープCZ-822C（本体）……………………￥69,800
- ●シャープCZ-881C（X1ターボZ）………………超特価☎
- ●シャープCZ-830(X1ツイン)＋カラーTVモニター…￥110,000
- ●シャープCZ-811C……………￥89,800⇨￥19,000
- ●シャープCZ-803C……………￥119,800⇨￥29,800
- ●シャープCZ-880C …………￥218,000⇨￥100,000
- ●シャープMZ-2861+1P-1252・￥383,000⇨￥245,000
- ●シャープMZ-5511…………￥　⇨￥45,000
- ●シャープMZ-5521 …………￥388,000⇨￥65,000
- ●シャープMZ-6551 …………￥430,000⇨超特価☎
- ●シャープMZ-6556…………￥650,000⇨超特価☎
- ●シャープMZ-2520…………￥159,800⇨￥78,000
- ●シャープMZ-2531…………￥198,000⇨￥120,000
- ●富士通FM77AV20-2 ………￥168,000⇨￥89,800
- ●富士通AV40EX……………￥168,000⇨￥126.000
- ●NEC PC-9801VX4 …………￥643,000⇨￥380,000
- ●NEC PC98XA2……………￥695,000⇨￥170,000

拡張機器他

- ●シャープCZ-8TM1(モデム)……￥29,800⇨￥9,800
- ●シャープMZ-1E29（RS232C カードケーブル付）￥17,800⇨￥9,800
- ●シャープX1用ジョイカード……………………￥1,500
- ●シャープCZ8EM(320KBボード)‥￥88,000⇨￥20,000
- ●シャープCZ-8EB-3（X1拡張 I/Oボックス）‥￥33,800⇨￥28,000
- ●シャープCZ-8EP(X1拡張ポート)‥￥11,800⇨￥9,000
- ●シャープMZ-1U01(2000用拡張)・￥37,000⇨（在庫切れ）
- ●シャープMZ-1U03(700用拡張)‥￥35,000⇨￥15,000
- ●シャープMZ-1U05(5500用拡張)‥￥12,000⇨￥8,500
- ●シャープMZ-1U09(2500用拡張)…￥9,000⇨￥7,200
- ●シャープ1R01+1R02×2………￥55,000⇨￥12,000
- ●シャープMZ-1E24232Cカード‥￥19,800⇨￥16,800
- ●シャープCZ-8BK3（第2水準 漢字ROM）…￥13,800⇨￥11,700
- ●シャープCZ-8BK4（第2水準 漢字ROM）……￥6,800⇨￥5,700
- ●シャープMZ-1M03（数値 プロセッサー）・￥69,000⇨￥35,000
- ●シャープMZ8BC04（GPIB ケーブル）……￥18,000⇨￥8,000
- ●シャープMZ-8BI04(GPIBカード)‥￥45,000⇨￥18,000
- ●シャープMZ-1R09（5500用 VRAM）……￥35,000⇨￥25,000
- ●シャープMZ-1R10（5500用 漢字ROM）…￥30,000⇨￥12,000
- ●シャープMZ-1R11（550用 256RAM）……￥80,000⇨￥40,000
- ●シャープMZ-1R18（1500RAM ファイル）…￥18,000⇨￥12,000
- ●シャープMZ-1R19（5500用 第二漢字ROM）・￥35,000⇨￥15,000
- ●シャープMZ-1R24(辞書ROM)…￥22,000⇨￥10,000
- ●シャープMZ-1R26A（増設RAM ボード）…￥15,000⇨￥12,800
- ●シャープMZ-1R27A（増設 ビデオRAM）‥￥13,000⇨￥10,000
- ●シャープMZ-1R28A（MZ-2500 辞書ROM）…￥13,000⇨￥10,000
- ●シャープMZ-1R29A（1P17第2 水準ROM）‥￥32,000⇨￥10,000
- ●シャープMZ-1R37（MZ-2500 RAMファイル）…￥35,800⇨￥28,000
- ●シャープMZ-1T02（MZ-2200 データレコーダー）￥19,800⇨￥8,500
- ●シャープMZ-1T03（MZ-5500用 データレコーダー）￥12,000⇨￥8,500
- ●シャープCZ8BGR2(X1ターボ10用)￥14,800⇨￥4,000
- ●シャープCZ-8BS1（ステレオ FM音源ボード）‥￥23,800⇨￥19,500
- ●シャープCZ-51F(ターボ増設)同等品………￥25,000
- ●シャープCZ-52F(X1F増設)同等品…………￥22,000
- ●シャープMZ-2000/2200/80B/1500/700用…………（フロッピー インターフェースカード）…………￥23,500⇨￥18,000
- ●シャープMZ-1E15（1.2MミニFD インターフェイス）・￥35,000⇨￥28,000
- ●シャープX1、MZ用マウス……………特価￥4,800
- ●シャープMZ-1X29(光学マウス)……￥13,800⇨￥11,000
- ●シャープMZ-1M08(ボイスボード)‥￥10,000⇨￥6,000
- ●シャープMZ-2000/22000キーボード………￥10,000
- ●シャープMZ-3500キーボード……………￥10,000
- ●シャープMZ-5500キーボード……………￥10,000
- ●シャープX1シリーズ用キーボード…………￥10,000

プリンター

- ●シャープMZ-1P27(水平プリンタ)‥￥268,000⇨￥214,400
- ●シャープMZ-1P28(80桁プリンタ)‥￥148,000⇨￥118,400
- ●シャープMZ-1P29(132桁プリンタ)・￥168,000⇨￥134,400
- ●シャープMZ-1P17（カラー漢字プリンタ ケーブル付）…………………￥85,800⇨￥39,800
- ●シャープMZ-6P11（1P10カットシート フィーダー）・￥95,000⇨￥35,000
- ●シャープCZ-8PD2(ドットプリンター)‥￥79,800⇨￥25,000
- ●シャープCZ-8PD3(ドットプリンター)‥￥59,800⇨￥19,800
- ●シャープCZ-8PK5(80桁)……￥129,000⇨￥69,800
- ●シャープCZ-8PK6(130桁)……￥159,000⇨￥89,800
- ●シャープCZ-8PC2（熱転写 プリンタ）……￥69,800⇨￥55,000
- ●シャープCZ-8NS1(イメージスキャナー)…………………￥188,000⇨
- ●シャープJX-100、200(カラースキャナー)　入荷予定！
- ●日立MP-1053（漢字プリンター 16インチ）‥￥315,000⇨￥158,000

フロッピーディスク

- ●シャープCZ-503F（5"2DX1インターフェースケーブル付）￥49,800⇨￥34,000
- ●シャープCZ-503（5"2DX1 インターフェースなし）…………￥30,000
- ●シャープCZ-502F（5"2DX1インターフェースケーブル付）￥99,800⇨￥75,000
- ●シャープCZ-300F…………………………￥13,000

ソフト

- ●シャープCZ-21 LLS(Cコンパイラ)…￥39,800⇨￥35,800
- ●シャープCZ141SF(NEW BASIC)￥18,800⇨￥16,000
- ●シャープMZ-2Z013（5500 MSDOS）…￥25,000⇨￥21,000
- ●シャープMZ-2Z017（5500 BASIC3）……￥20,000⇨￥17,000
- ●シャープMZ-2Z032（1500 DIKBASIC）…￥12,000⇨￥6,000
- ●シャープMZ-2Z064（MZ-6541用 新ワープロ）・￥69,800⇨￥59,500
- ●シャープMZ-1Z-005…………￥25,000⇨￥21,500
- ●シャープMZ-2Z023（MZ-5500 GWBASIC）‥￥50,000⇨￥42,500
- ●シャープMZ-2Z025（MZ-5500日本語ワープロ）￥49,800⇨￥15,000
- ●シャープMZ-2Z014(TODAY)…￥68,000⇨￥15,000
- ●シャープMZ5Z013(通信ソフト)……￥6,500⇨￥2,000
- ●シャープ6F03(QDディスク)……………… 10枚￥4,000
- ●シャープMZ-1E26（ボイスコミュニケーション）‥￥24,800⇨￥13,000
- ●シャープMZ-6Z010（テレフォン ソフトV2.0）…￥10,000⇨￥8,500
- ●シャープMZ-1M01(2000/2200用)………特価￥8,500

SHARPポケットコンピュータ

- ●PC1360(本体)…………………￥29,800⇨￥19,800
- ●PCE200(本体)…………………￥22,000⇨￥17.800
- ●PC-E500(本体)…………………￥28,800⇨￥24,800
- ●CE-150（カラーグラフィック プリンター）…………￥49,800⇨￥10,000
- ●CE-152(データレコーダ)……………￥19,800⇨￥9,800
- ●プログラムモジュール(CE-161)‥￥50,000⇨￥10,000
- ●プログラムモジュール(CE159)…￥35,000⇨￥6,500
- ●シャープCE-140Pカラープリンタ￥43,000⇨￥20,000

ポケコン総合カタログ並びに特価表を差し上げます。
切手￥70を同封の上、当社へお申込みください。

AIBIT
アイビット電子株式会社
〒192 東京都八王子市北野町560-5
☎0426-45-3001〜3
FAX.0426-44-6002
●営業時間:10:00〜19:00
●電話受付:20:00迄可
●定 休 日:日曜日(祭日営業)

※掲載されている商品は全て新品保証付きです。

信用をモットーに、よりよい品をより安く、迅速にお届けします。

全国通信販売
北海道から沖縄まで

★送料はご注文の際にお問い合わせ下さい。
★掲載の商品は、すべて新品、保証書付きです。
★掲載の商品は充分用意してありますが、ご注文の際は、在庫の確認の上、現金書留または、銀行振込でお申し込み下さい。全商品クレジットでも扱っております。
★お申し込みの際は必ず電話番号を明記して下さい。
★商品、品切れの節はご容赦下さい。

富士銀行八王子支店 （普）1752505

Let's CALL

超優良中古パソコンが電話一本で買える!! 03(797)1221

SHARP
CZ-811C (X-1F/10)
¥89,800➡ **¥12,000**

SHARP
CZ-812C (X-1F/20)
¥139,000➡ **¥42,000**

SHARP
CZ-820CE (X-1Gモデル10)
¥69,800➡ **¥16,800** 新品同様
X-1Gモデル10RFコンバータセット
(本体＋AN-58C) 新品同様
¥72,780➡ **¥19,600**
X-1Gモデル10ディスプレイセット
(本体＋CU-14GB) 新品同様
¥119,600➡ **¥46,600**

SHARP
CZ-822CB (X1-Gモデル30)
¥118,000➡ **¥59,800** 新品同様
X-1Gモデル30ディスプレイセット
(本体＋CU-14GB) 新品同様
¥167,800➡ **¥89,600**
X-1Gモデル30TVディスプレイセット
(本体＋TVディスプレイ)
¥197,800➡ **¥94,600**

SHARP
CZ-880CB 新品同様
(X-1Turbo Z本体)
¥218,000➡ **¥102,000**
CZ-880DB 新品同様
¥109,800➡ **¥86,000**
セット価格
¥327,800➡ **¥188,000**

SHARP
CZ-820DE・B 新品
(14インチ2000字RGBTV)
¥79,800➡ **¥39,800**

SHARP
CZ-8PK5 新品同様
(10インチ漢字プリンタ)
¥129,000➡ **¥69,800**
CZ-8PK6 新品同様
(15インチ漢字プリンタ)
¥159,000➡ **¥89,800**

SHARP
MZ-1P17(E・B) 新品
(色、グレー・ブラック)
(80桁カラー漢字熱転写プリンタ)
¥76,600➡ **¥42,800**
(X1用ケーブル付)
¥76,600➡ **¥46,800**
(MZ-2500用ケーブル付)

SHARP

本体

- CZ-811C (X-1F/10) ……… ¥89,000➡ **¥12,000**
- CZ-812C (X-1F model 20) ……… ¥139,800➡ **¥42,000**
- CZ-822C (X-1G model 30) ……… ¥118,000➡ **¥48,000**
- CZ-850C (X-1Turbo/model 10) ……… ¥168,000➡ **¥25,000**
- CZ-852C (X-1Turbo/model 30) ……… ¥248,000➡ **¥58,000**
- MZ-2531 (MZ-2500 V2) ……… ¥198,800➡ **¥88,000**

ディスプレイ

- 14M-111C (14"カラー1000文字) ……… ¥67,800➡ **¥15,000**
- CU-14F1 (14"カラー2000文字) ……… ¥64,800➡ **¥18,000**
- CU-14A2 (14"カラー4050文字) ……… ¥99,800➡ **¥48,000**
- CU-14AD (14"カラー4050文字) ……… ¥84,800➡ **¥48,000**
- CZ-811D (14"カラー2000文字RGBTV) ……… ¥89,800➡ **¥32,000**
- CZ-830D (14"カラー4050文字RGBTV) ……… ¥98,000➡ **¥62,000**
- MZ-1D05 (14"カラー2000文字) ……… ¥69,800➡ **¥18,000**
- MZ-1D11 (12"カラー4050文字) ……… ¥113,000➡ **¥42,000**
- MZ-1D22 (14"カラー2000文字) ……… ¥108,000➡ **¥45,000**

ディスクドライブ・プリンタ・他

- CZ-501F (5"2D、2ドライブ) ……… ¥129,800➡ **¥55,000**
- CZ-502F (5"2D、2ドライブ) ……… ¥99,800➡ **¥55,000**
- CZ-503F (5"2D、1ドライブ) ……… ¥49,800➡ **¥32,000**
- CZ-800P (10"ドットプリンタ) ……… ¥142,800➡ **¥15,000**
- CZ-81P (ミニサイズプリンタ) ……… ¥34,800➡ **¥10,000**
- CZ-8PP2 (カラープロッタプリンタ) 新品 ……… ¥54,800➡ **¥15,000**
- CZ-8PD2 (10"ドットプリンタ) ……… ¥79,800➡ **¥28,000**
- CZ-8PD3 (10"ドットプリンタ) ……… ¥59,800➡ **¥28,000**
- CZ-8PC1 (80桁24ドット漢字熱転写プリンタ) ……… ¥69,800➡ **¥35,000**
- CZ-8PN1 (80桁24ドット漢字熱転写プリンタ) ……… ¥134,800➡ **¥32,000**
- MZ-80P6 (80桁ドットプリンタ) ……… ¥155,500➡ **¥18,000**
- MZ-1P06 (80桁漢字プリンタ) ……… ¥234,000➡ **¥45,000**
- MZ-1P09 (MZ-1500用カラープロッタプリンタ) 新品 ……… ¥47,600➡ **¥25,800**
- MZ-1P17 (80桁24ドットカラー漢字熱転写プリンタ・CZ用ケーブル付) 新品 ……… ¥76,600➡ **¥42,800**
- MZ-1P17 (80桁24ドットカラー漢字熱転写プリンタ・MZ用ケーブル付) 新品 ……… ¥76,600➡ **¥46,800**
- CZ-8BM2 (X-1シリーズ用マウスインターフェイス) ……… ¥19,800➡ **¥10,000**
- CZ-8RB (X-1シリーズ用ROM BASIC) ……… ¥19,800➡ **¥10,000**
- CZ-8SS2 (システムスタンド) ……… ¥5,500➡ **¥4,000**

SHARP X-1シリーズ特選極上品コーナー

- CZ-820CE (X-1G/10) 新品同様 ……… ¥69,800➡ **¥16,800**
- CZ-822CB (X-1G/30) 新品同様 ……… ¥118,000➡ **¥59,800**
- CZ-880CB (X-1Turbo Z) 新品同様 ……… ¥218,000➡ **¥102,000**

SHARP ディスプレイ特選極上品コーナー

- CU-14G (14"カラー2000文字) 新品 ……… ¥49,800➡ **¥29,800**
- CU-14A4 (14"カラー4050文字) 新品 ……… ¥89,800➡ **¥49,800**
- CZ-820D (14"カラー2000文字RGBTV) 新品同様 ……… ¥79,800➡ **¥39,800**
- CZ-880DB (15"カラー4050文字RGBTV) 新品同様 ……… ¥109,800➡ **¥86,000**
- CZ-600D (15"カラー4050文字RGBTV) 新品同様 ……… ¥129,800➡ **¥88,000**

全商品保証付 6ヶ月の保証期間だから安心です。
クレジットでOK カレッジクレジットも取扱います。
全国無料配送 全国どこでも配達料はいただきません。
日曜配達可 留守の多い方でも安心です。
高額下取り 少ない予算で買いかえもラクラク。
高額買取り 電話1本で即、現金お支払い。
代金引換えシステム 商品到着時の代金支払いでOK。
ボーナス一括払い 商品は即お手元へ、お支払いはボーナス時に。

●電話一本で高額下取り、即商品はお手元へ!
●あなたの不要になったパソコンを電話一本で査定し買取ります。
●掲載の商品以外も取り扱っておりますのでお気軽にお電話下さい。

▼本社注文デスク
03(797)1221

コンピュータバンク

株式会社パシフィックコンピュータバンク 〒150 東京都渋谷区渋谷1-6-8 井上ビル 営業時間/AM9:30〜PM9:30 年中無休

できうる限りの誠意と責任をもって――IPL。

SHARP

SHARP X-68000

アクセス **No.X0805**

価格 ¥862,400 ➡ 超特価 **¥556,500**

CZ-600C(65536同時発色、ソフト〈グラデウス〉付、ステレオ音源)	¥369,000
CZ-601D(0.39ミリ、アナログ3モードオートスキャン)	¥119,800
VC-DS1000(Sデジタル・ノイズワイパー+S-VHS装備)	¥220,000
CZ-8VPI(ビデオマルチプロセッサ)	¥59,800
CZ-6VTI(カラーイメージユニット、テロッパー機能付き)	¥69,800
3Mブランクディスケット(5"2HD*10枚)	¥24,000
初期不良期間(ワイドに1ケ月間交換システム!)	¥0
安心の3倍保証(IPL保証書付き)	¥0

¥3,000×72回 ボーナス 4.02万×12回　標準価格¥862,400

¥5,000×72回	ボーナス 2.82万×12回
¥7,000×48回	ボーナス 3.88万×8回
¥10,000×36回	ボーナス 4.45万×6回

アクセス **No.X0801**

価格 ¥512,800 ➡ 超特価 **IPL超特価**

CZ-600C(65536同時発色、ソフト〈グラデウス〉付、ステレオ音源)	¥369,000
CZ-601D(0.39ミリ、アナログ3モードオートスキャン)	¥119,800
3Mブランクディスケット(5"2HD*10枚)	¥24,000
初期不良期間(ワイドに1ケ月間交換システム!)	¥0
安心の3倍保証(IPL保証書付き)	¥0

¥2.200×72回 ボーナス 2.0万×12回　標準価格¥512,800

¥3,000×48回	ボーナス 2.8万×8回
¥4,900×36回	ボーナス 3.0万×6回
¥10,000×24回	ボーナス 2.64万×4回

ARRANGE!
目的に合わせて
自由にアレンジ!!

X0801にゲームを添えて

アクセス **No.X0802**

CZ-6ST1(角度自由自在、調節OK!)	¥5,800
源平討魔伝	¥7,800
スペースハリアー	¥6,800
マンハッタン・レクイエム	¥7,800
CZ-8NJ1(ジョイカードプレゼント!)	¥0

価格 ¥541,400 ➡ 超特価 **IPL超特価**

¥2,700×72回　ボーナス 2.0万×12回

※その他のクレジットについてはお問い合せください。

X0801をシステムUP!

アクセス **No.X0803**

CZ-213MS(MUSIC PRO 68K)	¥18,800
CZ-214MS(SOUND PRO 68K)	¥15,800
CZ-8PC2(10"カラー熱転写B5～B4ハガキ可、全角半角文字)	¥69,800
A4カット紙(100枚)	¥470
CZ-8NJ1(ジョイカードプレゼント!)	¥0

価格 ¥617,670 ➡ 超特価 **IPL超特価**

¥3,000×72回　ボーナス 2.38万×12回

※その他のクレジットについてはお問い合せください。

50,000人もの人々が体感した安心感。
――信頼のIPLワイドサポート

アクセス **No.X0804**

価格 ¥587,800 ➡ 超特価 **¥365,800**

CZ-600C(65536同時発色、ソフト〈グラデウス〉付、ステレオ音源)	¥369,000
CZ-601D(0.39ミリ、アナログ3モードオートスキャン)	¥119,800
VC-N35(簡単操作のファミリービデオ)	¥75,000
3Mブランクディスケット(5"2HD 10枚)	¥24,000
初期不良期間(ワイドに1ケ月の交換システム!)	¥0
安心の3倍保証(IPL保証書付き)	¥0

¥3,000×72回 ボーナス 2.0万×12回　標準価格¥587,800

¥3,000×48回	ボーナス 3.5万×8回
¥6,000×36回	ボーナス 3.23万×6回
¥10,000×24回	ボーナス 3.98万×4回

SHARP X1 turbo Z

アクセス **No.X0817**

価格 ¥396,400 ➡ 超特価 **¥274,300**

CZ-881C(ビデオ画像取込、スーパーインポーズ、テロップ可、マウス付)	¥179,800
CZ-880D(14"カラーTVリモコン付4050文字)	¥109,800
VC-N35(簡単操作のファミリービデオ)	¥75,000
イースII	¥7,800
3Mブランクディスケット(5"2HD*10枚)	¥24,000
初期不良期間(ワイドに1ケ月の交換システム!)	¥0
安心の3倍保証(IPL保証書付き)	¥0

¥1,400×72回 ボーナス 2.0万×12回　標準価格¥396,400

¥3,000×48回	ボーナス 2.18万×8回
¥5,000×36回	ボーナス 2.13万×6回
¥9,100×24回	ボーナス 2.0万×4回

ARRANGE!
目的に合わせて
自由にアレンジ!!

X0817をS-VHSビデオに変えて

アクセス **No.X0818**

VC-DS1000(Sデジタル・ノイズワイパー+S-VHS装備)	¥220,000

価格 ¥541,400 ➡ 超特価 **¥358,000**

¥2,900×72回　ボーナス 2.0万×12回

※その他のクレジットについてはお問い合せください。

ボーナスフェア
実施7/18(MON)～8/18(THU)

Chance 1 期間中、システムお買い上げの方、先着200名様に、電話帳電卓をプレゼント。(電話番号・スケジュールを記憶、10桁電卓機能付)

Chance 2 期間中、デスクをお買上げの方全員に、A-300(原稿用スタンド¥8,000)をプレゼント

Chance 3 期間中、シャープ製品をシステムでお買上げの方全員にCZ-8NJ1(ジョイカード)をプレゼント

SHARP X68000 ACE

アクセス **No.X0811**

価格 ¥823,670 ➡ 超特価 **¥574,000**

CZ-601C(CPU68000、2Mバイト、65536同時発色)	¥319,800
CZ-601D(0.39ミリ、アナログ3モードオートスキャン)	¥119,800
VC-DS1000(Sデジタル・ノイズワイパー+S-VHS装備)	¥220,000
CZ-6VT1(カラーイメージユニット、テロッパー機能付き)	¥69,800
CZ-8PC2(10"カラー熱転写B5～B4ハガキ可、全角半角文字)	¥69,800
3Mブランクディスケット(5"2HD*10枚)	¥24,000
A4カット紙(100枚)	¥470
電話帳電卓*贈呈(電話番号50人分、スケジュールメモOK!電卓機能付)	¥0
初期不良期間(ワイドに1ケ月間交換システム!)	¥0
安心の3倍保証(IPL保証書付き)	¥0

¥3,000×72回 ボーナス 4.2万×12回　標準価格¥823,670

¥5,000×72回	ボーナス 3.0万×12回
¥8,000×48回	ボーナス 3.55万×8回
¥9,600×36回	ボーナス 5.0万×6回

◆全国無料配達◆

アクセス **No.X0812**

価格 ¥646,370 ➡ 超特価 **¥487,600**

CZ-601C(CPU68000、2Mバイト、65536同時発色)	¥319,800
CZ-601D(0.39ミリ、アナログ3モードオートスキャン)	¥119,800
CZ-6ST1(角度自由自在、調節OK!)	¥5,800
CZ-6BE1A(1MB増設RAMボード)	¥35,000
CZ-211LS(C compilerソフト開発を効率良く〈サポート〉)	¥39,800
XE-1 PRO(ジョイステック)	¥9,500
源平討魔伝	¥7,800
マンハッタン・レクイエム	¥7,800
スペースハリアー	¥6,800
CZ-8PC2(10"カラー熱転写B5～B4ハガキ可、全角半角文字)	¥69,800
3Mブランクディスケット(5"2HD*10枚)	¥24,000
A4カット紙(100枚)	¥470
電話帳電卓*贈呈(電話番号50人分、スケジュールメモOK!電卓機能付)	¥0
初期不良期間(ワイドに1ケ月間交換システム!)	¥0
安心の3倍保証(IPL保証書付き)	¥0

¥3,000×72回 ボーナス 3.3万×12回　標準価格¥646,370

¥5,000×72回	ボーナス 2.1万×12回
¥6,800×48回	ボーナス 3.0万×8回
¥10,000×36回	ボーナス 3.15万×6回

◆日曜・祭日指定配達◆

アクセス **No.X0813**

価格 ¥688,070 ➡ 超特価 **IPL超特価**

CZ-601C(CPU68000、2Mバイト、65536同時発色)	¥319,800
CZ-601D(0.39ミリ、アナログ3モードオートスキャン)	¥119,800
CZ-211LS(C compilerソフト開発を効率良く〈サポート〉)	¥39,800
CZ-213MS(MUSIC PRO 68K)	¥18,800
CZ-214MS(SOUND PRO 68K)	¥15,800
AN-160SP(アンプ内蔵スピーカーシステム2本組)	¥79,800
CZ-8PC2(10"カラー熱転写B5～B4ハガキ可、全角半角文字)	¥69,800
3Mブランクディスケット(5"2HD*10枚)	¥24,000
A4カット紙(100枚)	¥470
電話帳電卓*贈呈(電話番号50人分、スケジュールメモOK!電卓機能付)	¥0
初期不良期間(ワイドに1ケ月間交換システム!)	¥0
安心の3倍保証(IPL保証書付き)	¥0

¥3,000×72回 ボーナス 3.63万×12回　標準価格¥688,070

¥5,000×72回	ボーナス 2.42万×12回
¥7,000×48回	ボーナス 3.33万×8回
¥10,000×36回	ボーナス 3.72万×6回

超低金利………　組合せ自由　全国無料配送

※今回掲載の製品は、7月18日より8月18日までの期間に限らせていただきます。

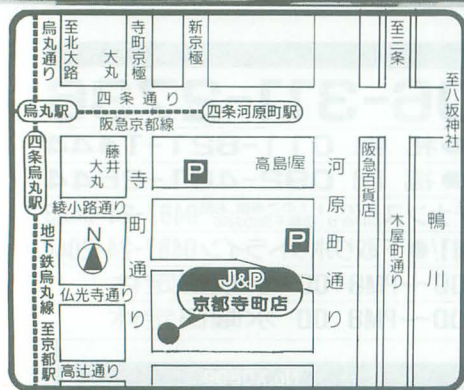
京都の人も、パソコンなら、

楽しさいっぱい京都寺町店のフロアあんない

3F 京都最大の中古品コーナー&セミナー
お目当ての機種が思わぬ価格で——お買得品満載の中古品コーナー。さらに、パソコンラックや専門書籍コーナーも充実。

2F ビジネスパソコンのフロア
32ビット・16ビットパソコンはもちろん、周辺機器、アクセサリーからビジネスソフトまで、何でも揃う情報戦士のためのフロア。話題のパソコン通信コーナーもご利用ください。

1F ワープロとホビーのフロア
ワープロ、PPC、FAXと、オフィスから書斎まで大活躍の電子文具のコーナーと、パソコンのもう一つの楽しみ、ゲームソフト、MSX、8ビットパソコンのホビーコーナー。見て回るだけでも楽しめます。

至北大路 烏丸通り　大丸　寺町京極　新京極　至三条　至八坂神社
四条通り　烏丸駅　四条河原町駅　阪急京都線
四条烏丸駅　藤井大丸　寺町通り　高島屋　河原町通り　阪急百貨店　木屋町通り　鴨川
綾小路通り　地下鉄烏丸線　仏光寺通り　至京都駅　高辻通り
J&P 京都寺町店

Joshin Computer Store
J&P 京都寺町店
京都市下京区寺町通仏光寺下ル恵美須之町549番(〒600)
☎(075)341-3571

常設無料セミナースケジュール(7月~8月)
※9月以降のスケジュールは受付にお問い合わせください。

日程	セミナーテーマ	時間
7/23(土)	財務管理/きよみずソフト「ワンタッチマスター」	PM2:00~3:00
	販売管理/OBC「TOP販売管理エキスパート」	PM2:00~3:00
7/24(日)	日本語ワープロ/管理工学研究所「新松」「桐」	PM2:00~4:00
7/30(土)	表計算/ロータス「Lotus 1-2-3」	PM2:00~4:00
7/31(日)	データベース/アスキー「The CARD2」	PM2:00~4:00
8/6(土)	英文ワープロ/高電社「KOA-Techno-Mate」	PM2:00~4:00
8/7(日)	日本語ワープロ/日英ワープロジャストシステム「duet」	PM2:00~4:00
8/13(土)	財務会計/PCA「PCA会計」	PM2:00~4:00
8/21(日)	財務会計/ミルキーウェイ「二代目大番頭」	PM2:00~4:00
8/27(土)	財務管理/きよみずソフト「ワンタッチマスター」	PM2:00~3:00
	財務会計/OBC「TOP財務会計エキスパート」	PM2:00~3:00
8/28(日)	ワープロソフト/dBソフト「コラージュ」	PM2:00~4:00

大阪の人も。J&Pです。

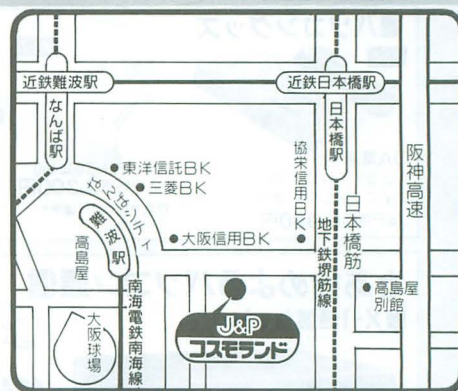
情報満載、コスモランドのイベントあんない

マッキントッシュフェア 4F
アップルマニア待望のアップルコーナー、アップルマッキントッシュフェアを開催。マックのソフトを多数揃えております。

ビジネスマンのためのフェアを連日開催 3F
有名ソフトハウスによるビジネスソフトフェア/OS.言語開催。さらに、毎週土・日には、株式に関するあらゆる情報が一堂にごらん頂ける通信による株価分析フェアと、J&Pホットライン・パソコン通信フェアを実施しています。

32・16ビット機に絞って充実の品揃え 2F
ビジネスパソコン、パソコンパーツ、プリンター、ハードディスク、各種周辺機器まで一堂に集結。NEC「新」PC-9801勢揃いフェアや、コンピュータ、ミュージック実演会も開催。

日本橋最大のワープロ売場誕生 1F
ワープロ売場では、ポータブルからデスクトップまで、日本橋最大の展示台数を誇ります。また、フルカラーコピーサービスもお気軽にご利用下さい。

サイズ	B5・A4	B4	A3
フルカラー	300円	400円	500円

大阪市浪速区難波中2-1-17
☎(06)634-3111

常設無料セミナースケジュール(7月)

※8月以降のスケジュールは受付にお問い合わせください。

日程	セミナーテーマ	時間
7/23(土)	POP作成ソフト/ソフトプロ「POP名人」	PM1:00〜4:00
	POP作成ソフト/ダイナウェア「PIX SPOT」	PM1:00〜4:00
7/24(日)	デスクトップパブリッシングソフト/モーリン「言図」	PM1:00〜3:00
	デスクトップパブリッシングソフト/dBソフト「コラージュ」	PM3:00〜5:00
	統合型ソフト/ヴァル研究所「ファラオII」	PM1:00〜2:00
	表計算/ロータス「ロータス1-2-3」	PM2:00〜3:00
7/30(土)	CADソフト/デザインオートメーション「CAD PAC98」	PM1:00〜3:00
	CADソフト/オートデスク「AUTO CAD」	PM3:00〜5:00
	株価分析/ダイツー「Z-Chart III」	PM1:00〜2:00
	グラフソフト/ダイナウェア「チャートUP」	PM2:00〜3:00
	データベース/ダイナウェア「UPクリッパー」	PM3:00〜4:00
7/31(日)	CADソフト/東京コンピュータ「Generic CAD」	PM1:00〜4:00

J&P オプション ソフト 通信販売

全国どこでも無料配達

パソコン通信

J&P HOT LINE でもお申し込みいただけます。

J&Pメールショッ

■シンプルで使いやすいパソコンラック・デスク・チェアー

X8-1 パソコンラック&チェアーセット
ラック寸法／幅600㎜3段棚
ラック：エレコムDS-10
チェアー：コイズミL-395
メーカー標準価格合計44,000円
セット特価 **23,000円**
●シートカラー ①青色 ②茶色

X8-2 パソコンシステムデスク
エレコムER-1200
J&P特価**29,000円**
幅1200×高さ650～1180 奥行750㎜

X8-3 エレコム PD-02
メーカー標準価格43,000円
J&P特価**19,800円**
コード落とし付
幅640%×高さ1305%×奥行700%

X8-4 エレコム PD-99＋FO-60E セット
メーカー標準価格合計51,500円
J&P特価**33,000円**
トレーユニット（FO-60E）をセットしてお得。
幅900%×高さ1280%×奥行700%

X8-5 パソコンチェアー
コイズミ L-395
キャスター付
メーカー標準価格12,000円
J&P特価**6,800円**
シートカラー ①青色 ②茶色

■パソコングッズ

X8-6 OA電源タップ
ナショナルWCH 4511
ノイズフィルター 集中スイッチ付
J&P特価**6,980円**

X8-7 エレコムSO-450
J&P特価**3,300円**
原稿が見やすく場所をとりません。

X8-8 5インチケース
100枚収納可
J&P特価**2,000円**

X8-9 3.5インチケース
80枚収納可
J&P特価**2,000円**

X8-10 PS-80
10インチプリンタスタンド
J&P価格**3,400円**
※プリンタ別売

■ポケコン

X8-11 PC-E200
J&P特価**17,800円**
Z80CPU採用で高速演算を実現。24桁4行表示

X8-12 PC-E500
J&P特価**24,800円**
充実の124関数機能、最大96Kバイトまで増設可能。40桁4行表示

さあ始めようパソコン通信

■X-1通信セット

X8-13
モデム：CZ-8TM2
J&P HOTLINE：スタータキット
通信速度300・1200bps
標準価格合計52,800円
セット価格**49,800円**

■X-1ターボ通信セット

X8-14
モデム：アイワ PV-A1200MKII 通信ソフト：SPS
JETターボターミナルJ&P
HOTLINE：スタータキット 通信速度300・1200bps
標準価格合計39,600円 セット価格**39,600円**

■電子手帳

X8-15
シャープPA-7000
J&P特価**17,800円**
これ1台で、電卓・電話帳・スケジュール・メモ・カレンダー機能があります。別売のモジュールを使うことにより、漢字辞書や英和・和英の翻訳機としても使えます。学生、技術者からビジネスマンまで幅広くお使いいただけます。

X8-16 ICカード（PA-7000用）

❶PA-7C1	英和・和英カード	6,300円
❷PA-7C2	漢字辞書カード	9,000円
❸PA-7C3	6ケ国語会話カード	6,300円
❹PA-7C4	カラオケ歌詞カード	9,000円
❺PA-7C10	電話帳・住所録カード	9,000円
❻PA-7C11	販売管理カード	9,000円
❼PA-7C12	技術計算カード	6,300円

X8-17 周辺機器

❶CE-152	データレコーダ	9,800円
❷CE-50P	プリンタ	17,800円
❸CE-200L	通信用ケーブル	2,500円

■〈X-1/ターボオプション〉

X8-18 マウス
シャープCZ-8NM2
J&P価格**6,800円**
X-1・MZ用マウス

X8-19 シャープCZ-8BV2
J&P価格**39,800円**
画像を自在に修正・加工できます
画像処理ツール・グラフィックソフト同梱

■X68000オプション X8-20

❶CZ-6BE1	1MB増設メモリ	35,000円
❷CZ-6BE2	2MB増設メモリ	79,800円
❸CZ-6BE4	4MB増設メモリ	138,000円
❹CZ-6BU1	ユニバーサルI/Oボード	39,800円
❺CZ-6BG1	GP-IBボード	59,800円
❻CZ-6BF1	RS-232C増設2チャンネル	49,800円
❼CZ-6BP1	68881数値演算プロセッサボード	79,800円
❽CZ-6EB1	拡張I/Oボックス4スロット	88,000円

■プリンタオプション X8-21

❶MZ-1C48	X-1シリーズ用プリンタケーブル	6,800円
❷MZ-1C35	MZ-2500/2200/2000用ケーブル	6,800円
❸MZ-1R29	MZ-1P17(B)用第2水準ROM	14,800円
❹CZ-8PC1-3	CZ-8PC1用第2水準ROM	9,800円

■ディスケット

X8-22 マクセル

❶MD2-D（10枚）	1,900円
❷MD2-DD（10枚）	2,400円
❸MD2-256HD（10枚）	2,500円
❹MF1-D（10枚）	4,500円
❺MF2-D（10枚）	4,500円
❻MF1-DD（10枚）	4,500円
❼MF2-DD（10枚）	5,000円
❽MF2-256HD（10枚）	8,300円

X8-23 J&Pオリジナル MD-2D（20枚） **3,000円**

X8-24 MD-2HD（10枚） **2,100円**

X8-25 MF-2DD（10枚） **4,000円**

全国無料配達

ピング

メールショッピングのお申し込みは J&P 渋谷店で承ります。

フロアーごあんない

- **4F パソコン教室** ●パソコン入門コース●BASIC上級コース ●BASIC初級コース●各種ビジネスコース
- **3F OA機器** ●ビジネスパソコン●ワードプロセッサ ●ビジネスソフト●OAサプライ ●ハンドヘルドコンピュータ
- **2F ビジネスパソコン** ●パソコン●ディスプレイ ●プリンタ●専門書籍 ●パソコンアクセサリー
- **1F ホビーのパソコン** ●ホビーパソコン●MSX ●ゲームソフト●学習ソフト

Personal Computer Store
J&P
渋谷店
東京都渋谷区道玄坂2丁目28番4号(〒150)
☎(03)496-4141〈水曜定休〉

■ビックヒットソフト

プロ野球ファン

注文No.	X8-26
適応機種	X-1シリーズ
ソフトハウス	日本テレネット

春の高校野球がスタートするまで冬眠でもしようと考えていた野球ファンのあなた。さあ、この真新しい球場で白球に賭けた男たちのドラマを味わってみてください。

¥7,800(5″2D)

ワールドゴルフⅡ

注文No.	X8-27
適応機種	X-1ターボ
ソフトハウス	日本テレネット

2モード(トレーニングモード、トーナメントモード)全72ホール構成。トーナメントモードでは、計100人のライバルゴルファーが登場。あらゆる角度からゴルフのおもしろさを徹底的に分析し、それをシミュレート。ゴルフゲームの最高峰作品!

¥7,800(5″2D)

注文No.	タイトル	ソフトハウス	適応機種	メディア	価格
X8-28	レリクス	ボーステック	X-1シリーズ	5″2D	¥7,500
X8-29	信長の野望(全国版)	光栄	X-1シリーズ	5″2D	¥9,800
X8-30	アルバトロス	日本テレネット	X-1シリーズ	5″2D	¥8,800
X8-31	殺意の接吻	リバーヒルソフト	X-1シリーズ X68000	5″2D	¥5,800
X8-32	棋太平	S・P・S	X-1シリーズ	5″2D	¥6,500

■Xホビーソフト

レジェンド

注文No.	X8-33
適応機種	X-1シリーズ
ソフトハウス	クエイザーソフト

人の心の光と闇を司るクリスタルを妖精アリーナが誤って地上に落してしまった。そのクリスタルを手に入れたのは古えの時代に神々をも滅ぼそうとした大魔王ガウティアであった。

¥7,800(5″2D)

蒼き狼と白き牝鹿ジンギスカン

注文No.	X8-34
適応機種	MZ-2500
ソフトハウス	光栄

「蒼き狼と白き牝鹿」の壮大なストーリーに加え、戦闘モードでは騎馬隊や弓矢隊など新しく加えられた戦闘部隊や略奪、狩猟、降伏勧告などの新コマンドも加わって、より複雑な戦略が楽しめるシミュレーションゲームとして期待できる。

¥9,800(3.5″DD)

注文No.	タイトル	ソフトハウス	適応機種	メディア	価格
X8-35	ウィザードリー3	アスキー	X-1ターボ	5″2D	¥9,800
X8-36	サジリ	日本テレネット	X-1ターボ	5″2D	¥7,800
X8-37	魔界復活	ソフトWING	X-1ターボ	5″2D	¥7,800
X8-38	ダ・ビンチ	HAL研究所	X1シリーズ	5″2D	¥6,800
X8-39	ディーヴァ	T&E	X1シリーズ	5″2D	¥7,800

■ビジネスソフト

注文No.	タイトル	ソフトハウス	適応機種	メディア	価格
X8-40	リバイバー	アルシスソフト	MZ-2500	3.5″DD	¥6,800
X8-41	ウィバーン	アルシスソフト	MZ-2500	3.5″DD	¥6,800
X8-42	殺人クラブ	リバーヒル	MZ-2500	3.5″DD	¥7,800
X8-43	ドルアーガの塔	テンバ	MZ-2500	3.5″DD	¥6,800
X8-44	スペースハリア	電波新聞社	X68000	5″2D	¥6,800

X-68000対応コーナー

Z'sSTAFF PRO 68K X8-45

表現力の素晴しさに加えて、編集機能もPRO仕様。複雑なカラーチェンジから、モザイク変換、ソフトフォーカスまで、じっくりと手の込んだ作品を描くことが可能である。

¥58,000 ●ソフトハウス(ツァイト)

X8-46

〈特長〉
- 一度に16個までウィンドウをオープンできます。
- マウス完全対応の簡単なオペレーション。
- Kamikaza(神風)はワープロ以上の表現力を持ちます。
- 簡単にデータをグラフ化することができます。

¥68,000 ●ソフトハウス(サムシンググッド)

X8-47
日本語ワープロ「将軍」

適応機種	X-1ターボ
ソフトハウス	シャープ

高性能ワープロソフト8ビットの最高峰

(5″2D) ¥34,800

X8-48
Inkpot

適応機種	X-1ターボ
ソフトハウス	アスキー

マウスでお絵かき、グラフィックソフト

(5″2D) ¥20,000

注文No.	タイトル	ソフトハウス	適応機種	メディア	価格
X8-49	C compiler	シャープ	X68000	5″2D	¥39,800
X8-50	MUSIC	シャープ	X68000	5″2D	¥15,800
X8-51	サウンドPRO 68K	シャープ	X68000	5″2D	¥15,800
X8-52	LINKS 68K	シャープ	X68000	5″2D	¥19,800
X8-53	日本語MY CARD・X1t	アバロン	X68000	5″2D(2)	¥58,000

お申し込み方法

右の注文書にご希望商品の注文No.および必要事項ご記入の上、現金書留にて J&P 渋谷店までお申し込みください。現金受領後、発送いたします。
また、J&P HOTLINE会員の方は、ショッピングコーナーでもお申し込みいただけます。

●記載以外のパーツのご注文も承ります。
詳しくはお電話にてお問い合わせ下さい。
☎(03)496-4141 定休:毎週水曜日

キリトリ線

現金書留申込み用紙

おところ 〒□□□-□□

TEL ( )

おなまえ 様

注文No.	数量	金額
X8-( )		円
X8-( )		円
合計		円

お手持ちのパソコン

お申込み先:東京都渋谷区道玄坂2丁目28番4号(〒150) J&P 渋谷店メールショッピング係

ACCESS

＊プロテクトの施してあるソフトは実行できません
＊一部サポートしていない機能があります
＊原理上実行できないソフトもございます

X1エミュレータはX1シリーズのアプリケーションソフトをX68000上で実行して頂くためのソフトウェアエミュレータです。X1のアプリケーションを完全にソフトウェアのみでエミュレートしているため、実行速度は平均3～5倍程度遅くなりますが、今までX1上でお使い頂いていたアプリケーションがHuman68k上でお使い頂けます。

X68000ではX1ソフト(5"2D)のメディアを取り扱うことができませんので、付属の専用ケーブルを接続してX1ソフトをHuman68kのディスク上にファイル転送し、そのファイルを参照してエミュレートを行ないます。Human68k上に仮想的にX1のドライブを作りますので、X1で使用しているイメージのままお使い頂けます。

X1とX68000間のファイル転送用ユーティリティを用意しておりますのでたんにファイルコンバータとしてもお使い頂けます。

X1シリーズ用実行可能アプリケーションソフト

- BASIC
- LISP
- PASCAL
- CP/M
- COBOL
- C
- XILOGO
- FORTRAN
- FORTH……etc

X1エミュレータ

定価¥9,800

異機種ソフトを利用

MS-DOSエミュレータ CONCERTO-X68K

定価¥99,800

CONCERTO-X68K(コンチェルト)は、X68000上でMS-DOSのアプリケーションをお使い頂くためのMS-DOSエミュレータです。NEC V30CPUを使用した専用ハードウェア(DOS Engine)が付属しており、ハードウェアによる高速実行を実現しています。

MS-DOSソフトのDOS Engine上での実行の管理、およびそれからのコールを専用エミュレーションソフトがサポートし、特定機種用と限定されていないMS-DOS(Ver2.11)用のソフトがX68000上でお使い頂けます。また、MS-DOS(Ver2.11)をお持ちの方は、それに付属のCOMMAND. COMを起動することによりMS-DOS上で作業しているのと同じイメージで、つまりX68000を疑似的にMS-DOSマシンとして使用することができます。CONCERTO-X68KはX68000の世界をより一層広げることをお約束致します。

DOS Engine

(V30 CPUボード)

MS-DOS用実行可能アプリケーションソフト

- MS-C(Ver4.00)
- MS-FORTRAN(Ver3.13, 4.01)
- MS-PASCAL(Ver3.13)
- MS-LINK(Ver2.01, 2.20, 2.44)
- MS-BASIC(Ver5.27)
- Lattice C(Ver2.12, 3.10)
- Qputimizing-C(Ver2.20F)
- TURBO PASCAL(Ver2.00B, 3.01A)
- Plink86(Verl.46)
- etc………

(特　長)

- 8MHzのV30を使用
- メモリは512KByte搭載
- オプションで8087NDP実装可能

＊ボードは本体より12cm程度大きくなります。その部分にはカバーがつきます。

代理店募集

アクセスではこれらの製品の発売にあたり代理店を募集しております。詳しくはお問い合せください。

＊MS-DOSはマイクロソフト社, CP/Mはデジタルリサーチ社の商標です。
COMMAND.COMはMS-DOSに標準のコマンドプロセッサです。上記のソフトウェアは各社の商標です。
＊製品の仕様、名称は予告なく変更する場合もございますのであらかじめご了承ください。

有限会社アクセス　〒101 東京都千代田区神田神保町1-64
神保町協和ビル7F
☎03(233)0200(代)　FAX.03(291)7019

ワープロでつかまえた、夏。

夏をホントに楽しみたいなら、ぜひ1台ワープロを用意してホシイ。旅に、料理に、株式にと、使って便利な情報が、指先一つで手に入るから。タイピングなんて知らなくていい。データをフロッピーに貯めこめば、編集・印刷だって思いのまま。自分専用の情報がアッと言う間に組み立てられる。ワープロ通信なら、「使えるネット」のJ&P HOT LINE。この夏、休みは知的にすごしたい。

ご入会はスタータキットで

夏の旅行は、HOT LINEにおまかせ。

■どこへ行こうか？行き先もワープロで調べます。／
●旅行情報●日本のまつり●映画(東京・大阪)●コンサート
■もう、時刻表はいりません？／●交通情報(新幹線・航空時刻表)
■泊まり先だって、HOT LINEで探せます。／ホテルガイド

暑中見舞いもエレクトリカルに！

■通信仲間みんなに一気に送信！／電子メール(グループ送信等多機能)
■手紙もワープロで書く時代です。／ワープロ文例集
■仲間づくりも簡単に！／独自の構成のBBSと多彩なSIG

夏をのりきるスタミナ料理。

■SHARP・サンヨー・日立、各社の知恵を絞って／電子レンジ教室
■経済料理ならおまかせ！／ワンポイントクッキング
■食の話題もいろいろあります！／生活情報「遊ing」

ボーナスは、賢く増やして、上手に使う。

■上場企業の動きを毎日お届け！／野村証券情報(会社ニュース)
■株価を分析、手がたくかせごう！／CUG「Zチャート研究会」
＊専用ソフトが必要です。
■お家にいながらにして、お贈りものを！／オンラインショッピング(中元商品)

アクセスポイント全国89ヵ所!!

1200bps/300bpsサポートポイント
東京・大阪・名古屋/札幌・苫小牧/青森山形・新潟・仙台・水戸・土浦・大宮・鹿島・立川・船橋・千葉・川崎・横浜・横須賀・平塚・甲府・静岡・金沢・京都・神戸・岡山・松江・広島/徳島・高松/福岡・長崎・鹿児島

300bpsサポートポイント 旭川・函館/八戸・秋田・盛岡・米沢・福島・郡山・いわき・宇都宮・前橋・高崎・太田・熊谷八王子・長野・松本・上田・諏訪・沼津・浜松・富山・高岡・石川・福井・岐阜・豊橋・大垣・津・四日市・大津・奈良・堺・貝塚・和歌山・尼崎・姫路・米子・福山・津山・呉・山口・徳山・下関・宇部/新居浜・松山・高知/北九州・佐賀・久留米・熊本佐世保・大分・宮崎/浦添

■申込先
〒556 大阪市浪速区日本橋5-6-7
上新電機株式会社
J&P HOT LINE 事務局
TEL. (06) 632-2521

■利用料金について
入会金/3,000円(スタータキット購入の代金から充当されます。)
接続料/3分あたり20円(電話料金はお近くのアクセスポイントまででOKです。)

スタータキット申込書

お名前		お電話番号	(　　) ー
ご住所	〒		
お申込品	スタータキット(ソフトなし) ¥3,000		

●パソコン通信ネットワークサービス

J&P HOT LINE

▼万全のサポート体制で全国をネットするパソコンの大型専門店 J&P チェーン

店名	住所	電話
渋谷店	東京都渋谷区道玄坂2丁目28番4号	☎(03) 496-4141
町田店	東京都町田市森野1丁目39番16号	☎(0427)23-1313
八王子店	東京都八王子市旭町1番1号八王子そごう7F	☎(0426)26-4141
テクノランド	大阪市浪速区日本橋5丁目6番7号	☎(06) 634-1211
メディアランド	大阪市浪速区日本橋5丁目8番26号	☎(06) 634-1511
コスモランド	大阪市浪速区難波中2丁目1番17号	☎(06) 634-3111
ワープロランド	大阪市浪速区日本橋4丁目9番15号	☎(06) 634-1411
ビジネスランド	大阪市北区梅田1-1-3大阪駅前第3ビルB2	☎(06) 348-1881
阪急三番街店	大阪市北区芝田1-1-3 阪急三番街B1	☎(06) 374-3311
高槻店	高槻市高槻町11番16号	☎(0726)85-1212
くずは店	枚方市楠葉花園町15番2号	☎(0720)56-8181
千里中央店	豊中市新千里東町1-3-204千里サンタウン3F	☎(06) 834-4141
摂津富田店	高槻市大畑町24-10	☎(0726)93-7521
寝屋川店	寝屋川市緑町4-20	☎(0720)34-1166
藤井寺店	藤井寺市岡2丁目1番33号	☎(0729)38-2111
岸和田店	岸和田市土生町2451-3	☎(0724)37-1021
京都寺町店	京都市下京区寺町通仏光寺下ル恵美須之町549	☎(075)341-3571
京都近鉄店	京都市下京区烏丸通七条下ル東塩小路町702	☎(075)341-5769
姫路店	姫路市東延末1丁目1番住友生命姫路南ビル1F	☎(0792)22-1221
和歌山店	和歌山市元寺町4丁目4番地	☎(0734)28-1441

※通信可能のワープロ機種についてはお近くのJ&Pまで、お気軽にご相談ください。

T4910217908548 雑誌 02179-8